

# CURRENT NOTES

The Newsletter for ATARI Users of Maryland, D.C. and Northern Virginia

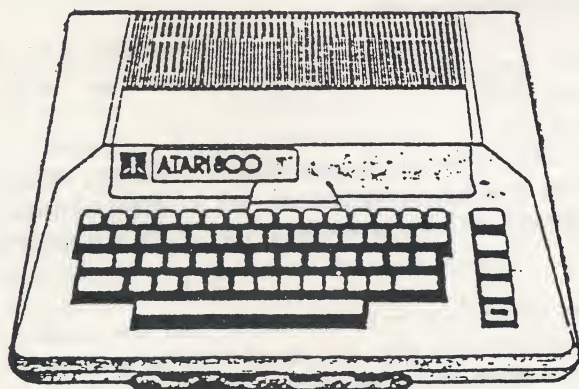
Volume 4, Number 2  
February, 1984

## In This Issue



In this month's issue of CURRENT NOTES, you will find many interesting and educational articles written by the members of our Atari user groups. You will find Joe Waters' program BLOCKADE which will be the topic of Novataris February BASIC tutorial. Also to be found is Jim Campbell's monthly article on ATARI LOGO. He introduces us to LOGO and presents two sample programs. Joe Waters presents the second part of his series BASIC BEAT. This month he continues to explain how to write a disk directory program. Bruce Blake talks about disk drives in his article DISK DRIVE INTERFACE (via SIO). This is followed by Bob Kelly's article ATARI SCUTTLEBITS. This month he compares the PCjr. with the ATARI 800XL. Some great Atari secrets are revealed in the article SUNNYVALE SECRETS written by the Secret Sunnyvale Correspondent. Preppie II and Lode Runner are reviewed this month in Jay Gerber's software review column NIBBLES AND BITS. Bill Shadt makes programming a little easier with his utility FINDIT. And last, but not least, Mike Barrett tells us about a new service provided by the Tysons Corner Center in his article COMPUTER BROWSING.

All this, as well as information about club meetings, libraries, and the BBS, is found in this issue of CURRENT NOTES. Happy reading !!



## INSIDE

Article	Author	page
In This Issue	Staffan Sandberg	FC
Editors Notes	Staffan Sandberg	2
Club Officers		3
Where and When		3
Novataris Notes	Joe Waters	4
DC Currents	Arthur Corte	4
AURA Update	Bruce McLendon	5
AURA Minutes	Rochelle Follender	5
DC Library News	Bob Danson	6
New Passwords!	John Lauer	6
A.N.A.L.O.G. Update	John Brophy	7
Armudic Update	John Brophy	7
Dear Editor		7
BLOCKADE	Joe Waters	8
Atari LOGO	Jim Campbell	10
BASIC Beat	Joe Waters	12
Disk Drive Interface	Bruce Blake	20
Atari Scuttlebits	Bob Kelly	22
Upcoming Events		23
Nibbles and Bits	Jay Gerber	24
TID BITS		25
Sunnyvale Secrets	Secret Sunnyvale Corr.	26
Tech Talk	John Baum	26
Find It	Bill Shadt	27
Computer Browsing	Mike Barrett	27
Index to Advertisers		27

Editors Notes

by Staffan Sandberg, Editor

Wow! Putting together this months issue of CURRENT NOTES was quite a job. I would like to thank all the writers for their time and effort. If you would like to write for the newsletter, please contact me at home or at one of the meetings. If you have a modem, the new Current Notes Automated Editor will be online from 9 PM Friday through 7 AM Saturday and 9 PM Saturday through 7 AM Sunday starting January 27. This is a BBS. Please contact me for a password.

To make the job of being editor a little easier, I have split the position into five jobs. These are editor, membership director, advertising director, promotional director, and correspondence director. John Lauer is membership director and I am editor. If you are interested in any of the other positions, please contact me. Thank you and happy Valentine's Day!

## CONSUMER ELECTRONICS

**PRESENTS DISCOUNT PRICES**

PERCOM AT-88-S-1 ATARI SINGLE DRIVE.....\$329.00  
 PERCOM AT-88-A-1 ATARI ADD ON DISK DRIVE S/S...\$263.00  
 PERCOM AT-88-S1 PD WITH PRINTER PORT S/S.....\$460.00

GENINI-10X PRINTER \$289.00

AMDEK COLOR 1 MONITOR 13".....\$285.00  
 AMDEK GREEN SCREEN 12".....\$145.00

THIRD PARTY SOFTWARE - YOU SELECT THE DISCOUNT!!  
 ASK ABOUT AVAILABILITY AND PRICE BY MFG. AND TITLE

15% OFF RETAIL ON ALL ORDERS

20% OFF RETAIL ON ALL ORDERS OVER \$100.00 AT RETAIL  
 25% OFF RETAIL ON ALL ORDERS OVER \$200.00 AT RETAIL  
 30% OFF RETAIL ON ALL ORDERS OVER \$300.00 AT RETAIL

ALL ORDERS SHIPPED UPS - ADD 3% SHIPPING AND HANDLING  
 - MINIMUM SHIPPING CHARGE \$3.00. MARYLAND RESIDENTS  
 ADD 5% SALES TAX. ALL ORDERS MUST BE ACCOMPANIED BY A  
 CASHIERS CHECK, MONEY ORDER OR A PERSONAL CHECK MADE  
 PAYABLE TO: CONSUMER ELECTRONICS. A PERSONAL CHECK  
 WILL DELAY YOUR ORDER FOR 15 BUSINESS DAYS. SEND YOUR  
 ORDERS OR FOR MORE INFORMATION SEND A SELF-ADDRESSED  
 STAMPED ENVELOPE TO:

CONSUMER ELECTRONICS  
 19213 DEEP RUN COURT  
 GERMANTOWN, MD 20874  
 ATTN: ORDER DEPARTMENT

THANK YOU FOR SAVING WITH CONSUMER ELECTRONICS !

CURRENT NOTES

CURRENT NOTES is published monthly free of charge to the members of the Atari Club of downtown DC, Novatari (the Northern Virginia Atari Users Group), and A.U.R.A. (the Atari Regional Association of Maryland) by Current Notes, 11804 Magruder Lane, Rockville, Maryland 20852. Second-class postage paid at Rockville, MD.

The three member groups are independent groups for Atari computer users, and are not affiliated in any way with Atari, Inc.

The Editor of CURRENT NOTES is Staffan Sandberg, 11804 Magruder Lane, Rockville, Maryland 20852. CURRENT NOTES telephone number is 301-468-6686. News items, short articles, original programs, classified ads, and any other material of interest to the membership are eagerly solicited. The deadline for articles is the 2nd Friday of the preceeding month.

Membership dues for both groups are \$15.00 a year, which includes subscription to CURRENT NOTES. Dues are payable at the beginning of each calendar year. Dues for new members joining during the year are reduced \$1.00 for each month which has passed since the first of the year. Dues may be paid at any meeting, or be sent to the editor. Persons living outside the metropolitan Washington DC area may subscribe to CURRENT NOTES for \$12.00 per year.

Advertising policy: classified ads are free to members. Commercial advertising Rate Cards are available upon request.



CLUB OFFICERSCurrent Notes

Staffan Sandberg	Editor	(301) 468-6686
Jim Cambell	Staff Writer	(703) 425-1440
Jay Gerber	Staff Writer	(703) 525-9715
Bob Kelly	Staff Writer	(301) 839-7377
Secret Sunnyvale Corr.	Staff Writer	
Joe Waters	Staff Writer	(703) 430-1215
Mike Barrett	Contributer	(703) 437-7522
Bruce Blake	Contributer	(301) 599-8888
Bill Shadt	Contributer	
Joe Waters	Contributer	(703) 430-1215

Armodic BBS

John Brophy	SYSOP	(703) 425-7169
	BBS	(703) 425-6698
John Lauer	Passwords	(301) 599-1621

DC Users Group

Frank Huband	President	(202) 527-4770
Fred Stollnitz	Treasurer	(301) 681-5748
Gerald Whitmore	Membership	(301) 459-6164
Arthur B Corte	Program Chairman	(703) 437-7860
Jim Cambell	V Program Chairman	(703) 425-1440
Bruce Ingalls	Tape Librarian	(703) 430-3287
Bob Danson	Disk Librarian	(703) 780-0758
John Brophy	ANALOG Disk	(703) 425-7169

Novatari Users Group

Frank Potter	President	(703) 225-4225
Steve Steinberg	Vice President	(703) 435-2962
Joe Waters	Program Chairman	(703) 430-1215
Curtis Sandler	Treasurer	(703) 734-9533
Tim Kilby	Secretary	(703) 987-8054
M. Evan Brooks	Disk Librarian	(703) 354-4482

AURA Users Group

Bruce McLendon	President	(301) 587-7890
Dave Haseman	Vice President	(301) 681-5776
David Curry	Treasurer	
Rochelle Follender	Recording Secretary	
Richard Stoll	Membership	(301) 946-8435
No Candidate	Corresponding Secretary	
No Candidate	SYSOP	

AURA GROUP MEETINGS

are held on the first Wednesday of every month at 7 PM in Room One of the Long Branch Public Library on Garland Avenue in East Silver Spring. Take the Beltway (I-495) to Exit 29-B South (University Blvd East, Route 193). Follow University Blvd. East (Route 193) to the second light (Piney Branch Rd.). Turn right on Piney Branch Rd. and continue to the second light (Arliss St). Turn right on Arliss St. past the apartments to Garland Avenue. Turn right on Garland Ave. The Long Branch Library is on the corner. Park in the Library's lot. Due to construction, please use the upper-level entrance.

NOVATARI MEETINGS

are on the second Sunday of each month. Novatari meets in the Greenbriar Community Center on Stringfellow Road in Chantilly, Virginia. Stringfellow Road, also known as Route 645, runs south from US 50 a little more than two miles west of the Fair Oaks Shopping Mall (intersection of I-66 and 50). There is a traffic light where Stringfellow Road meets route 50. The Greenbriar Community Center is on the left-hand side of Stringfellow Road, 1.4 miles south of 50. There is a small parking lot in front and a larger one just north of the center (that is, just before you get to the center). The meeting room is available from 5-9 PM. We offer a BASIC tutorial from 6:00 to 6:30 each month. We also offer a monthly arcade tournament that begins at 6:30. The business meeting starts at 7:00 and is followed by two formal half-hour presentations, one focusing on hardware and the other on software.

DC GROUP MEETINGS

are held on the third Tuesday of every month in Room 543 of the National Science Foundation offices, 1800 G Street NW, Washington, DC. The closest subway stop is Farragut West, on the Blue and Orange Lines. Take the 18th Street exit, and walk south (against the flow of traffic) down 18th Street for 3 blocks to G Street. The building is on the corner of 18th and G; it can be identified by a sign for the Madison National Bank on the corner. Front entrance is in the middle of the block. Parking is available in the building, for a fee. The entrance is on the west side of 18th Street, between F and G. Meetings begin at 5:30 PM and usually last until 8 or 9.



## NOVATARI NOTES

### February 12: New Disk Drives and the Turtle

Which Drive? As many of you are aware, ATARI owners now have quite a selection of new disk drives to choose from. ATARI has introduced the 1050 as a replacement to the 510. But there will be plenty of competition from other manufacturers including PERCOM, TRAK, RANA, ASTRA, INDUS. AMDEK has even introduced a 3" microfloppy disk drive for the ATARI. Several members will be available to relate their experience with one or more of these new drives. We will also try to have as many on hand for observation as we can get.

ATARI LOGO. Along with new disk drives, we also have several new languages to choose from, one of the most exciting of which is ATARI LOGO. Recent reviews ("ATARI LOGO Son of LISP," by Brian Moriarty, A.N.A.L.O.G. no. 14, 1984 and "ATARI LOGO: Looking Good," by Ken Harms, ANTIC, September, 1983) have strongly endorsed the ATARI version of LOGO. If you are looking for a language that is very easy to learn, powerful, and fun, ATARI LOGO may very well be your answer. Jim Campbell will be our featured speaker on LOGO. Jim, who will author the Current Notes LOGO column, presented LOGO to the DC Group in January and is looking forward to using his notes once again for us.

ARCHON Tournament. Kids, practice your swordsmanship, laser blasts, and spell-casting. We will have an ARCHON tournament this month. Since it may take awhile to eliminate some opponents, we will try to have two computers available for the tournament. At the January meeting, Dan Greenblatt (35,735) and Joe Hertz (32,625) were the top two players for JUMPMAN. Craig Heatwole (30,350) and Joe Waters Jr. (26,500) led the contest in MINER 2049er.

January Meeting. In spite of a scheduling mix-up which delayed the start of our meeting until 6:30, the new format of the NOVATARI meetings was well-received. The election of "new" officers resulted in the President and Vice President switching places (Frank Potter is now President and Steve Steinberg Vice President). Tim Kilby remains Secretary and Curtis Sandler Treasurer.

A motion was made to authorize the new Vice President to spend \$300 on the bulk purchase of diskettes and a more limited sum on tape cassettes. The diskettes and tapes will be made available to members at cost at the February meeting.

Joe Waters started his BASIC Tutorial by introducing a simple program called BLOCKADE (listed separately in this issue of CURRENT NOTES) and discussing some of the available books and magazines on the ATARI and ATARI BASIC. The game will serve as the focal point of the tutorial over

the next few months.

A special thanks to John Baum of STS Video, 1073 West Broad Street, Fall Church, VA. (237-0558), for an excellent presentation on maintenance procedures and the new ATARI Service Contracts. Audience questions and interaction with the speaker clearly indicated a keen interest in the topic. Thanks also to Joe Waters for his discussion of the functions and capabilities of ATARI DOS. Because we ran out of time, the discussion of OSS A+ had to be abbreviated. We'll squeeze it in again at some future meeting.

Coming Events. March: Software: the OSS ACTION language; Hardware: the 600XL and 800XL computers. April: Software: Database packages; Hardware: the ATARI printers. May: Software: new Music programs; Hardware: the new 1400XL and 1450XL computers (if they are available). June: Software: Graphics packages; Hardware: Joysticks, Trakballs, etc.

## DC CURRENTS

### February 21: THE TAXMAN COMETH

As the unpleasant task of completing Form 1040 approaches, the time is ripe to review what our Ataris can do to aid us in this painful exercise in slimming our checkbooks. Having demonstrated the "Home Accountant" from Continental Software, in January, we will proceed to its companion program "Tax Advantage" which uses the data files of the Home Accountant as part of its input.

The meeting will also feature a talk by John Baum of STS Video, our local Atari service center on the care and feeding of our Ataris to keep them in proper running order and answer questions about what problems we can expect and what to do about them. His January presentation to the Novatari group was well received and this will allow the D.C. group to benefit from John's expertise.

Finally, since we want our meetings to be fun, we will play a bit with the "Pinball Construction Set" by Bill Budge, advertized with the following quote attributed to Steve Wozniak "The best program ever written for an 8-bit machine"



AURA\_UPDATE

LONGBRANCH PUBLIC LIBRARY - TAKOMA PARK, MD

AGENDA for FEBRUARY 1, 1984

AURA MINUTES - January 4, 1984

- 7:00 PM Informal discussions and disk/cassette claims
- 7:30 PM CONVENE MEETING  
 Reading of the minutes 1/4/84  
 Treasurer's report  
 Communications report  
 Changes in the newsletter - present & future
- 7:45 PM ANNOUNCEMENTS  
 Next meeting date, place and time  
 ELECTIONS & DUES TONIGHT - \$15!  
 (No newsletters without payment of dues!)  
 Announcements from the membership  
 OFFICER VACANCIES:  
     SYStem OPERator  
     Treasurer  
     Corresponding Secretary
- 8:00 PM FIRST PRESENTATION  
 'The Wedge' is a clever utility that allows you a limited extension of BASIC, such a viewing variables, etc. After only a few months, Steve Gauss has some interesting additions he would like share with us. The Wedge is public domain and is in our library!
- 8:15 PM SECOND PRESENTATION  
 A spectacular World War I aerial dog-fighting game, BLUE MAX, will be demonstrated by Mike Rinzel. Test your skill and learn something about flying as well!
- 8:30 PM OPEN FLOOR  
 GENERAL ELECTIONS and PAYMENT OF DUES!!  
 Formal announcement of additional committees needing staff and peripheral support.  
 Additional A.U.R.A. Officer vacancies.  
 General questions and interchange of information from the membership and, if present, selected guest speakers.
- 8:45 PM ADJOURNMENT of GENERAL BUSINESS MEETING

1. Next meeting will be February 1 at 7:00.
2. AURA part of the COMPUTER AGE BBS will be discontinued due to lack of use. In near future we hope to have an answering machine on Applied Computer Associates' (ACA) 340-0100 number which, after normal business hours will give an AURA message after the regular business message telling about the next meeting agenda, snow emergency, etc. Eventually, the 424-4112 number will be a BBS.
3. Current issue of "ANTIC" has printer-related articles, including one on how to print fonts. Current issue of "Creative Computing" has an article on how to write off your computer on your taxes.
4. For those who have an 800XL or 1200XL, ATARI has a free translator program that will run Atari 800 programs. Have your serial number and call ATARI at (800) 538-8543.
5. Staffan Sandberg, editor of "Current Notes" described future plans for the newsletter--it will be at least 24 pages with never more than 25% ads. Articles are wanted and can be downloaded from a modem. Call him at 468-6686.
6. Linc Halen demonstrated SCITOR, a personal finance and record keeping package. It is menu driven, interfaces with VISICALC, and you can have up to 60 catagories. He rated it 'very good' for a small business.
7. Bob demo'd the 800XL and the RANA drive. The disc drive has had some quality control problems, but otherwise okay. XL has nice features. To get fine scrolling, POKE 622,255 (not described in manual).
8. Bruce demo'd the TRAK drive. Has a printer port built in and a 4K buffer among its features.
9. The group voted to have dues of \$15.00 per year with no pro-rating as of this time.

Due to construction on the upper-level entrance to the library, we may no longer use this entrance. Therefore, we must exit DOWNSTAIRS as the library closes - Everyone must be out of the building by 9:00 PM! Please help maintain our good standing with the library staff by leaving by 9 PM!

D.C. Library News

by Bob Danson

New DCOM:

This month brings a complete revision to one of the Groups most popular volumes, the Data Communications disk. DCOM has been expanded and replaced by a two-disk set with optional printed documentation. JTERM has been replaced with a modified version (3.8) that will run at 300 or 1200 baud. AMODEM has been updated to include a new version (1.82) and a binary load version. DISKFER, a program to transfer entire disks, has been added. Many of the documentation text files have been revised and a new file describing all the disk files has been added. The two-disk set will be available for \$7. For those members with the old DCOM disk the new disks can be obtained for \$3 and the original DCOM disk in trade. If you don't have a printer, listings of the text files (17 pages) can be obtained for an additional \$1.

Hidden Option:

There is a "hidden" option in the D.C. menu program that is not shown in the Help screen. By entering a single numeric digit that corresponds to a disk drive address (such as "1" for D1) the directory from the disk in that drive will be read. This allows you to change disks without re-running the program.

DUP.SYS 2.6f:

The modified version of DUP.SYS provided on the DFIX volume will not work properly if it is written to another disk using it's "H" command. This problem, apparently an interaction between DUP.SYS and the menu program on the DFIX volume, can be circumvented by using the standard ATARI DUP.SYS to format a new disk and using the "H" command to save the standard ATARI DOS to that disk. Then, using the "C" or "O" DUP.SYS command, copy DUP.SYS 2.6f from the DFIX volume to the new disk in place of the standard DUP.SYS. DUP.SYS 2.6f can then be used normally from the new disk.

Programs Needed!

Please! If you have any public domain programs that are not in the D.C. Library contribute them! If you're not sure if a program is in the public domain, contribute it anyway - all programs are screened prior to publication. Bring your programs to any D.C. meeting or call Bob Danson at 780-0758.

New D.C. Disk Librarian Wanted

The time has come for a new D.C. Disk Librarian. I took on the task of being D.C. Librarian one year ago and now wish to move on to other activities within the Group.

If you have a bit (a lot?) of time to spend on managing, maintaining, and creating the D.C. Library, desire to learn more about your ATARI hardware and software, and want to work with other interesting and knowledgeable ATARI users, this is the job for you. It'll be hard work, but extremely rewarding! Call Bob Danson at 780-0758 for additional information.

NEW PASSWORDS!!

by John Lauer

New member numbers and passwords have been issued this month. They can be found on the top row of your mailing label. Please note that your old member number and password have now been voided.





## A.N.A.L.O.G. Disk Library

by John Brophy

We are setting up a group purchase of the A.N.A.L.O.G. Compendium. There is no discount on the price, but the clubs will get a free set of disks. We intend to distribute the disks to Compendium purchasers at our cost for the media, less any price break we can get on shipping charges.

To participate, send \$16.95 (checks payable to John Brophy) to:

John Brophy  
A.N.A.L.O.G. Librarian  
9300 Shari Drive  
Fairfax, Va. 22032

Be sure to include your NAME, ADDRESS, AND PHONE NUMBER, so I can contact you when the books arrive. Unidentifiable money will be donated to Children's Hospital. Delivery will be made at the Novatari and DC meetings, or by appointment at my house.

The deadline is 11 Feb 84; any money received after that will be returned. I would appreciate getting orders as soon as possible, since the more I have when I call the magazine, the better our bargaining position will be.

## ARMUDIC (?) Update

by John Brophy, SYSOP

STATISTICS: During 15-17 January, we received 244 calls including 34 disconnects, for an average of 70 calls/day. Average call length was 10.5 minutes, which means the line was in use 12 hours, 18 minutes per day. Median call length was 9 minutes, most frequent length was 5 minutes.

(N)EWS--19 Jan: This column will be available as menu option N; it will be updated more frequently than the newsletter, I hope, & the date it was written will appear in the menu. I expect a lot of readers initially, who were looking for the (B)BS Numbers, which will become option B.

(D)ownload problems--Please tell me what software you are running when you have difficulty. Remember, ARMUDIC does NOT support XMODEM protocol. If you are uploading, use LISTED files; put ML into a BASIC program that POKES decimal values.

ARMUDIC?--was named for its original phone #. A pronounceable word for the new number might be a good idea. I have listed all combinations, and the sayable ones start with GAL, HAL, or HAK, followed by M, N, or O, & end in NXT, MYT, NYT, NXT, or OXT. Or perhaps some genius can come up

with a catchy sentence whose initials spell the number. Vote for your favorite with option L, and I'll print the totals in (N)EWS.

## DEAR EDITOR

December 28, 1983

Dear Staffan,

The short program ("Text in Graphics 8") by Chuck Delo reprinted in the December newsletter does a good job of placing text on the 640x480 hi-res graphics screen. It lets you write text anywhere on the screen, permits sub- and superscripting, and even features double-height letters (12 lines per screen). It compares favorably to "T:", the POX autorun utility that lets you print text and graphics in any graphics mode. The trouble with "T:" is that, like so many Atari utilities, it boots from a cold start and therefore can't be used with other autorun programs.

Delo's program as printed had one minor "feature," however, that disturbed me: in double-height mode the vertical strokes appear as dotted lines. Only every other scan line "lights up."

Here's a short patch for those who prefer their letters solid. It replaces line 9910 of the existing program.

```
9910 D3=D1+YY*TT+XX+U-1:FOR Z=0 TO 7:POKE
D3+Z*TT,PEEK(D2+Z)
9920 IF TT=80 THEN D4=D1+(YY+0.5)*TT+XX+U-1:POKE
D4+Z*TT,PEEK(D2+Z)
9930 NEXT Z:NEXT U:RETURN
```

Also enclosed is a check for membership renewal. There's much more to our twin user groups than the newsletter, but "Current Notes" alone is worth the membership. Congratulations to both you and Paul Chasin on a great job.

Sincerely,



Bud Stoiker



**BLOCKADE**

by Joe Waters

Way back in August, 1981, COMPUTE! published a short BASIC game for the ATARI called BLOCKADE developed by Douglas Pinho. The idea for this version is a direct descendent of that early COMPUTE! article although even back then the game concept used in BLOCKADE was not new. This is a two-player game. Each player uses a joystick to control the direction of movement of a growing line. The idea is to keep your line growing as long as possible without running into (1) the border, (2) your opponent's line, or (3) your own line. Since your opponent cannot run into your line, you can, with clever placement of your own line, block his/her access to whole areas of the playing field.

I choose this game to start our tutorial series for several reasons. The game itself is very simple to understand and requires a minimum of physical dexterity. Players who have never even held a joystick soon find that they can put on a competitive performance. The "competitive" aspect was another reason. This is a great party game. After just a few rounds, you will see Grandpa and Uncle George locked in a deadly serious battle to see just who will block who! Besides the playability aspects, the program is not too long and it does illustrate the use of a great many of the commands found in ATARI BASIC.

Since we will be using this program to illustrate ATARI BASIC, I will not explain the whole thing in detail now but rather just give a brief sketch of the main parts. Lines 20 - 80 initialize a number of variables. In most cases these variables are used to improve the readability of the program (GOSUB QUIET is easier to understand than GOSUB 172). Various short subroutines used by the program are defined in the lines between 100 and 300.

The main program starts at line 310 and ends at 350. We draw a title page (subroutine TITLE is in lines 905-980), draw the playing field (subroutine FIELD, 705-845), begin the game (subroutine BEGIN, 610-680), draw the lines until someone crashes (subroutine ACTION, 405-499). After a crash, if neither player has reached a score of 10, we go back to draw another field and play another round. When one of the players reaches a score of 10, we display the winner (subroutine DONE, 505-585) and ask the players if they want another game. If yes, we go back to TITLE where one of the other game options can be chosen if desired. If no, game is over.

Try and type the game in and play it before the February meeting. Be sure to bring your copy to the tutorial that will start at 6:00. For those of you who have a modem, I will try and put the game on ARMUDIC. If you

don't yet have a "text" book, I would recommend Inside ATARI Basic: A Fast, Fun, and Friendly Approach, by Bill Carris as a painless introduction to the language for adults and kids alike. One of the best all-around guides to the ATARI computer (including the program recorder, disk drives, and printers) as well as ATARI BASIC is Your ATARI Computer: a guide to ATARI 400/800 Personal Computers, by Ion Pool with Martin McNiff & Steven Cook.

```

1 REM =====
2 REM !                               B L O C A D E
!
4 REM !                               Joe Waters
!
6 REM !Novatari Tutorial
February, 1984 !
10 REM ===== INITIALIZE VARIABLES
=====
20 =53279:START=6:OPTION=3
40 DIM GMS(3):GMS(1)=121:GMS(2)=96:GMS(3)=81
50 DIM B$(40),BLANK$(40):BLANK$(1)=" ":BLANK$(40)="
":BLANK$(2)=BLANK$(1)
60 180:BOX=200:BLOCKS=220:PAUSE=240
70 IELD=700:TITLE=900
80 GOTO MAIN
100 REM ===== JOY: READ JOYSTICKS
=====
106 J=STICK(PN-1)
108 IF J=15 OR J=5 OR J=6 OR J=9 OR J=10 THEN DX=0:DY=0:GOTO
122
110 IF J=7 THEN DX=1:DY=0:GOTO 118
112 IF J=11 THEN DX=-1:DY=0:GOTO 118
114 IF J=13 THEN DX=0:DY=1:GOTO 118
116 IF J=14 THEN DX=0:DY=-1
118 IF PN=1 THEN DX1=DX:DY1=DY:GOTO 122
120 IF PN=2 THEN DX2=DX:DY2=DY
122 RETURN
130 REM ===== PR: PRINT TEXT SCREEN
=====
136 POKE TXTROW,VAL(B$(1,1)):POKE TXTCOL,VAL(B$(2,3))
140 PRINT B$(4,LEN(B$)):RETURN
150 REM ===== SCORE: SHOW SCORE
=====
156 POKE TXTROW,2:POKE TXTCOL,4:?" P1S:" ";
158 POKE TXTROW,2:POKE TXTCOL,34:?" P2S:" ";
160 RETURN
170 REM ===== QUIET: SHUT SOUNDS OFF
=====
176 FOR S=0 TO 3:SOUND S,0,0,0:NEXT S:RETURN
180 REM ===== BELL: MAKE BELL SOUND
=====

```



```

86 FOR V=14 TO 0 STEP -.4
188 SOUND 0,PITCH,10,V:SOUND 1,PITCH+1,10,V
190 NEXT V:GOSUB QUIET:RETURN
200 REM ===== BOX: DRAW SMALL BOX
=====
206 PLOT X,Y:DRAWTO X+1,Y:DRAWTO X+1,Y+1:DRAWTO X,Y+1:DRAWTO
X,Y
208 GOSUB QUIET:RETURN
220 REM ===== BLOCKS: ADD TO GAME 3
=====
226 COLOR 3:X3=INT(XMAX/3)-5:FOR I=1 TO 2*ROUND
232 X=RND(0)*X3:Y=RND(0)*YMAX:SOUND 0,X+Y,10,8:GOSUB BOX
234 X=X1+4+RND(0)*X3:Y=RND(0)*YMAX:SOUND 1,X+Y,10,8:GOSUB
BOX
236 X=X2+4+RND(0)*X3:Y=RND(0)*YMAX:SOUND 2,X+Y,10,8:GOSUB
BOX
238 NEXT I:GOSUB QUIET:RETURN
240 REM ===== PAUSE: WAIT A WHILE
=====
242 FOR W=1 TO WAIT:NEXT W:RETURN
300 REM =====> MAIN: PROGRAM STARTS HERE
(<=====
310 GOSUB TITLE
320 GOSUB FIELD
330 GOSUB BEGIN
340 GOSUB ACTION
350 IF P15<10 AND P25<10 THEN GOTO 320
360 GOSUB DONE
370 IF B$="Y" THEN P15=0:P25=0:ROUND=0:GOTO 310
380 END
390 REM ===== PROGRAM ENDS HERE
=====
400 REM ===== ACTION: MOVE PLAYERS UNTIL CRASH
=====
405 PN=1:GOSUB JOY:X1=X1+DX1:Y1=Y1+DY1
410 PN=2:GOSUB JOY:X2=X2+DX2:Y2=Y2+DY2
415 COLOR 1:LOCATE X1,Y1,Z1:PLOT X1,Y1:SOUND 1,X1+Y1,10,10
420 COLOR 2:LOCATE X2,Y2,Z2:PLOT X2,Y2:SOUND 2,X2+Y2,10,10
425 IF X1=X2 AND Y1=Y2 THEN Z1=2:Z2=1:GOTO 445:REM BOTH
CRASH
430 IF Z1<>0 OR Z2<>0 THEN GOTO 445:REM ONE PLAYER CRASHES
435 GOSUB QUIET:GOTO 405:REM NOBODY CRASHES. KEEP GOING.
440 REM -----> CRASH DETECTED
<-----
445 IF Z1>0 THEN P15=P15+1:TONE=60
450 IF Z2>0 THEN P25=P25+1:TONE=243
452 IF Z1>0 AND Z2>0 THEN TONE=121
455 GOSUB QUIET
460 FOR L=0 TO 14:SOUND 0,200-10*L,10,L:GOSUB 490:NEXT L
465 FOR W=1 TO 5:PITCH=10*W+RND(0)*10+100:FOR V=14 TO 0 STEP
-.2
470 SOUND 0,PITCH,10,V:SOUND 1,PITCH+1,10,V:NEXT V:NEXT
W:GOSUB PAUSE
472 PITCH=TONE:GOSUB SCORE:GOSUB BELL:GOSUB PAUSE
475 FOR L=14 TO 0 STEP -.6:SOUND 0,10*L,10,L:GOSUB 490:NEXT
L

```

```

485 WAIT=200:GOSUB PAUSE:RETURN
489 REM CHANGE COLOR OF LOSING PLAYER
490 IF Z1>0 THEN SETCOLOR 0,P1C,L
495 IF Z2>0 THEN SETCOLOR 1,P2C,L
499 RETURN
500 REM ===== DONE: GAME FINISHED, SHOW WINNER
=====
505 GRAPHICS 2+16:IF P15+P25>=20 THEN GOTO 555
510 IF P15+P25>=20 THEN GOTO 555
515 POSITION 7,2:? #6;"PLAYER";:POSITION 8,5
520 IF P15=10 THEN B$="TWO"
525 IF P25=10 THEN B$="ONE"
527 ? #6;B$:POSITION 3,8:? #6;"is the winner!":WAIT=50
530 FOR I=1 TO 6
535 POSITION 8,5:? #6;" "":GOSUB PAUSE
540 POSITION 8,5:? #6;B$:GOSUB PAUSE
545 NEXT I:GOTO 570
550 FOR I=1 TO 6
555 POSITION 5,5:? #6;"GAME TIED!":GOSUB PAUSE
560 POSITION 5,5:? #6;"game tied!":GOSUB PAUSE
565 NEXT I
570 POSITION 3,10:? #6;"another game?";
575 OPEN #1,4,0,"K":GET #1,X:B$=CHR$(X):CLOSE #1
580 IF B$="Y" OR B$="N" THEN POSITION 17,10:? #6;B$:RETURN
585 SOUND 0,X,12,10:FOR W=1 TO 20:NEXT W:SOUND 0,0,0,0:GOTO
575
600 REM ===== BEGIN: GET PLAYERS READY
=====
610 =ROUND+1
620 B$="017ROUND":GOSUB PR
630 =" "":GOSUB PR
640 WAIT=100:GOSUB PAUSE
650 B$="217READY":GOSUB PR:PITCH=121:GOSUB BELL
660 B$="217 SET "":GOSUB PR:PITCH=108:GOSUB BELL
670 B$="217 GO! "":GOSUB PR:PITCH=96:GOSUB BELL
680 RETURN
700 REM ===== FIELD: BORDER AND INITIAL POSITIONS
=====
705 GRAPHICS 6M:POKE CRSINH,1
710 P1C=INT(RND(0)*16)
715 P2C=INT(RND(0)*16):IF P2C=P1C THEN GOTO 715
720 BRC=INT(RND(0)*16):IF BRC=P1C OR BRC=P2C THEN GOTO 720
725 SETCOLOR 0,P1C,4:SETCOLOR 1,P2C,8:SETCOLOR
2,BRC,4:SETCOLOR 4,0,0
730 COLOR 3:REM DRAW BORDER IN COLOR NO. 3
735 PLOT 0,0:DRAWTO XMAX,0:DRAWTO XMAX,YMAX:DRAWTO
0,YMAX:DRAWTO 0,0
740 REM ==> PICK START POSITIONS <==
745 X1=INT(XMAX/3):X2=2*X1
750 +RND(0)*Y3)
755 COLOR 1:PLOT X1,Y1:REM DRAW PLAYER 1
760 COLOR 2:PLOT X2,Y2:REM DRAW PLAYER 2
765 IF OPT=3 THEN GOSUB BLOCKS:REM MAKE GAME 3 HARDER
770 B$="001PLAYER 1":GOSUB PR:B$="031PLAYER 2":GOSUB PR
780 B$="102SCORE":GOSUB PR:B$="132SCORE":GOSUB PR
790 IF ROUND>0 THEN GOTO 810

```



**ATARI LOGO**

by Jim Campbell

```

795 REM ==> ONLY PRINT THIS MESSAGE FOR THE FIRST ROUND <==
800 B$="012First Player to":GOSUB PR
805 B$="112Score 10 Loses!":GOSUB PR
810 GOSUB SCORE
815 B$="307(PRESS BUTTON TO CONTINUE)":GOSUB PR
820 IF STRIG(0)<>0 AND STRIG(1)<>0 THEN GOTO 820:REM WAIT
FOR BUTTON
825 REM ==> ERASE 1ST MESSAGE <==
830 B$="012":GOSUB PR
835 B$="112":GOSUB PR
840 B$="307":GOSUB PR
845 RETURN
900 REM ===== TITLE: WRITE TITLE PAGE
=====
905 GRAPHICS 2:SETCOLOR 2,0,0:POKE CRSINH,1:OPT=1
910 POSITION 6,3:? #6;"blockade"
915 POSITION 7,5:? #6;"GAME ";OPT
920 B$="007USE OPTION TO CHANGE GAME.":GOSUB PR
925 B$="210USE START TO BEGIN.":GOSUB PR
930 REM ==> CHECK CONSOLE KEYS FOR OPTION OR START <==
935 POKE CONSOLE,8:X=PEEK(CONSOLE):IF X<>START AND X<>OPTION
THEN GOTO 935
940 IF X=START THEN GOTO 960
945 OPT=OPT+1:IF OPT=4 THEN OPT=1
950 POSITION 12,5:? #6;OPT:PITCH=6MS(OPT):GOSUB BELL:GOTO
935
955 REM ==> SET GRAPHICS MODE, X AND Y LIMITS <==
960 IF OPT=1 THEN GM=3:XMAX=39:YMAX=19
965 IF OPT=2 THEN GM=5:XMAX=79:YMAX=39
970 IF OPT=3 THEN GM=7:XMAX=159:YMAX=79
975 P1S=0:P2S=0:ROUND=0:REM START WITH SCORES=0
980 RETURN

```

**RENUMBER-IT  
IN A FLASH**

An Atari BASIC line renumbering utility,  
**RENUMBER-IT** offers the following  
features:

- 100% Machine Language
- Does Not Use Page Six
- Co-Resident-Callable At Any Time
- Renumbers Internal Line References
- Lists Lines With Variable Line References
- User Friendly

All these features for only \$9.95 plus  
plus \$2.00 shipping and handling. Order your's  
now from **SAGEWARE SYSTEMS**.  
DISK Only.MC, VISA, Check or M.O.

**SAGEWARE SYSTEMS**

P.O. BOX 45, Owasso, OK 74055  
918/272-3428

What was the underlying reason you decided to buy your computer? Price, games, lots of software, popularity, or a super-slick sales person? No matter, but how much thought was given to the programming aspects of the computer?

Remember when you bought Pac-Man and the salesman reached up on the shelf and said "You'll be needing the BASIC cartridge if you're going to do any programming"? After two weeks of playing Pac-Man, you began to wonder what else the computer could do. Your kids asked you to write a BASIC program, so you created a program to display their names on the screen. You then asked your twelve-year-old to do something--anything--besides Pac-Man, and I'll bet you got every excuse in the book.

In the data processing environment there is a saying: "K.I.S.S.," which means "Keep It Simple, Stupid." Your family can learn computer language if you work to achieve only a little success at a time. Many day-to-day problems are resolved by breaking them down into smaller parts and then solving the small parts or problems. Why not treat computer questions in the same manner?

Logo is a great language because it encourages you and your children to explore use of the language by moving "turtles" around the screen. You'll want to try making your own screen designs and varying the collrs. While doing this, you'll make mistakes, which is another way of learning. The Logo package will prompt you out of any problem; weird results may display when you first try your procedure, but you can quickly correct the commands and execute the procedure again. Your attention is focused on one procedure at a time, but on e procedure may call another, and then another. You execute one procedure, it looks good, then several in a small sequence, and they check out. Results are instantaneous, and it is a great feeling when all procedures execute correctly.

The following Logo program, called "Start," was written by Brian Harvey of Atari, and demonstrates the power of procedures. Draw your own screen design by moving the "joystick" and depressing the trigger occasionally. The other program, called "Plants," is easier to follow but requires the use of an arc (curves).

Reverence materials are quickly becoming available and I recommend the following books from the local public library:

Mindstorms - Papert Seymour

Apple Logo - Harold Abelson/McGraw-Hill (90% compatible to Atari Logo)

Atari Logo - Can be ordered from Atari by calling 800-538-8543, costs approximately \$99.95



```

TO START
INSTRUCTIONS
SETUP
HELP
DRAW 5
END

```

```

TO INSTRUCTIONS
SETBG 1
TS CT SETCURSOR [10 2]
PR [VIDEO TURTLE]
PR [] PR []
PR [PLUG IN A JOYSTICK IN PORT #1]
PR []
PR [TO CHANGE TURTLE'S PENCOLOR:]
PR [PRESS THE JOYSTICK BUTTON]
PR []
PR [TURTLE HAS YELLOW PEN DOWN]
PR [] PR []
PR [TO RECALL COMMANDS:]
PR [PRESS H FOR HELP]
SETCURSOR [4 20]
PR [* PRESS ANY KEY TO BEGIN *]
KEYPRESS RC
END

```

```

TO KEYPRESS :ANYKEY
END

```

```

TO CHANGEPN
IF PN=2 [SETPN 0 STOP]
SETPN ( PN + 1 )
END

```

```

TO CHANGETURT.CLR
IF PN = 0 [SETC 15]
IF PN = 1 [SETC 75]
IF PN = 2 [SETC 35]
END

```

```

TO DRAW :STEP
IF KEYP [PEN.UP.DOWN RC]
CHECKJOY JOY 0
DRAW :STEP
END

```

```

TO SETUP
CT CS FS ST TELL 0
SETPN 0
SETPC 1 75
SETPC 2 35
SETBG 1
SETC 15
WHEN 3 [CHANGEPN CHANGETURT.CLR]
END

```

```

TO HELP
CT SS
PR [U PICKS UP THE PEN]
PR [D PUTS DOWN THE PEN]
PR [X PUTS REVERSE PEN DOWN]
PR [C CLEARS THE SCREEN]
PR [E ERASE HELP]
END

```

```

TO PEN.UP.DOWN :UPDOWN
IF :UPDOWN = "E [CT FS]
IF :UPDOWN = "H [HELP]
IF :UPDOWN = "X [PX]
IF :UPDOWN = "C [SETUP]
IF :UPDOWN = "U [PU]
IF :UPDOWN = "D [PD]
END

```

```

TO CHECKJOY :POS
IF :POS < 0 [STOP]
SETH 45 * :POS
FD :STEP
FD :STEP
END

```

```

TO PLANTS :DIST
FS
PENUP
LT 90
FD :DIST
RT 90
PENDOWN
PLANT FD 90
PENUP
RT 90
FD :DIST * 2
LT 90
PENDOWN
SETPN 2
PLANT
SETPN 0
END

```

```

TO PLANT
PENUP
FD 50
PENDOWN
FLOWER
BACK 120
PETAL
BACK 20
END

```

```

TO FLOWER
REPEAT 10 [PETAL RT 360 / 10]
END

```

```

TO PETAL
ARC 50 90
RT 90
ARC 50 90
RT 90
END

```

```

TO ARC1 :STEP :TIMES
REPEAT :TIMES [RT 5 FD :STEP RT 5]
END

```

```

TO ARC :RADIUS :DEG
ARC1 0.174532 * :RADIUS :DEG / 10
IF 0 = REMAINDER :DEG 10 [STOP]
FD 0.174532 * :RADIUS / 10 * REMAINDER :DEG 10
RT REMAINDER :DEG 10
END

```



## BASIC BEAT

by Joe Waters

### Review of the Problem.

Last month, to help illustrate BASIC programming techniques, we started a project to solve a common problem: the management of our diskette library. Since one of the most fundamental precepts of programming is to understand what we want to accomplish, let's restate our ultimate goal. We want to see, either on the screen or printed on paper, the answers to questions that we have about our diskette library. In addition, we want to obtain our answers with minimal effort on our part.

Now, what will it take to accomplish this? To answer questions about our disk library, we need information about our library, or in other words, a database filled with data. Before we fill anything with data, we have to decide what data we want and how to store it. Before we store the data, we have to get it. For this problem, that means we have to be able to read the disk directory and save that information in a file. The directory is on the diskette in one of our disk drives. Therefore, before we can even get started, we have to know something about the number and kinds of disk drives attached to our system. Similarly, when we get ready to print results, we will have to know something about the characteristics of our printer.

We could get the information we need by asking the user about the working environment everytime the program is run. However, our normal working environment may not change much and answering the same set of questions all the time would soon become tedious. Alternatively, we could specify the relevant characteristics right in our program. This is usually the easiest solution, but it does have disadvantages. First of all, the program would only work for our system. And if we did make any changes (we want our printed listings a little wider or we purchase a new disk drive), we would have to search out the relevant variables in the original code and change them -- sometimes much easier said than done.

In this month's column, we illustrate a more general solution to the problem. We create a file to store information about the working environment. When the program starts, it reads this file and sets all relevant parameters accordingly. An additional option in our main menu allows us to change, and save if desired, any of these parameters any time we want.

### Outline of a Solution.

When the program starts, it looks for the file that contains data on our working environment. If the file is found, it is read and control passes to the main menu. If the file is not found, the user is prompted for the relevant

facts and the file is created before going on to the main menu. The main menu is expanded to include an option to change any of these settings.

We need information on disk drives and the printer. We also allow the user to customize screen color. To handle disk drives, we introduce an eight-element array called DTYPE (for Drive TYPE). Each of the eight elements corresponds to one of the possible eight disk drives that can be linked to an ATARI. If the value of an element is zero, the drive is not available. A value of 1 indicates a single density drive; 2 indicates double density. The setting of printer options is postponed until we are ready to deal with printing. Color options are indicated by three variables: BKC (background color), BKL (background luminance), and CHL (character luminance).

### Implementation of the Solution.

Sounds easy enough. Now, how do we implement the solution? Change the main program to call a routine, named PROFILE, before going to MENU. Develop separate routines to read the profile data (RPROFILE), save the data (SPROFILE), create the original data (CPROFILE), and change the profile (CHPRO). The changes are accomplished in three distinct routines: CHDISK for the disk drives, CHPRINT for the printer, and CHCOLOR for colors. Several short utility routines are also introduced as we go along.

### Methodology.

Before launching into code development, let me explain the method of presentation. Last month all of the code was printed at the end of the article. I don't want to duplicate the entire program each month as it grows. At the same time, listing fragments at the end of the text would impede the tutorial nature of the effort. So I have decided to list the various components we are adding (or revising) right with the descriptive text. If you add these lines to your program, you have a working copy of the current state of the effort. However, many of the individual components can stand on their own and be incorporated into your other programs.

### Main Program

The only change in our main program is the addition of a line (225) calling the PROFILE subroutine:

```

200 REM =====
201 REM > MAIN PROGRAM STARTS HERE <
202 REM =====
210 GOSUB INIT
220 GOSUB TITLE
225 GOSUB PROFILE
230 GOSUB MENU
240 END
250 REM =====

```



### New Subroutines

I include the subroutine names at the beginning of the program. The CHPRD routine is put at line 9000 following the other main menu routines. The KEYBD routine was introduced last month. This month we add four additional short utility routines (line 110). The routines relating to the PROFILE are in line 120. Routines changing the settings are listed in line 130.

```
90 LIBRD=5000:LIBUP=6000:LIBSR=7000
100 LIBPR=8000:CHPRD=9000
110 KEYBD=300:SCREEN=340:BADKEY=350:PA
USE=360:PRCEN=370
120 PROFILE=11000:CPROFILE=11200:RPROF
ILE=11300:SPROFILE=11400:HPROFILE=11600
130 CHDISK=13000:CHPRINT=14000:CHCOLOR=
15000:RDCOL=15200:RDLUM=15300
```

### New Variables

Recall that the initial values of our variables and the dimensions of arrays are found in the INIT routine. B\$ is used as a general string variable. BLANK\$ holds all blanks. I've already mentioned the DTYPE array. Note that the initial setting assumes all zeros except for a 1 in DTYPE(1), i.e. one single density disk drive. DSTAT\$ holds the English equivalent of the 0,1,2 values found in DTYPE. The four variables UP\$, DN\$, LT\$, and RT\$ hold the four arrow keys on the ATARI. I've defined them using the CHR\$ function only because most printers can't print these arrow characters.

```
19000 REM *****
19001 REM INIT: INITIALIZE VARIABLES
19002 REM *****
19010 DIM K$(1),B$(40),BLANK$(38)
19020 DIM DTYPE(8),DSTAT$(42)
19030 DIM UP$(2),DN$(2),LT$(2),RT$(2)
19100 B$(1)=" ":B$(40)=" ":B$(2)=B$(1)
19102 BLANK$(1)=" ":BLANK$(38)=" ":BLA
NK$(2)=BLANK$(1)
19105 BKC=9:BKL=4:CHL=10
19110 FOR I=1 TO 8:DTYPE(I)=0:NEXT I:D
TYPE(1)=1
19120 DSTAT$="NOT AVAILABLESINGLE DEN
SITYDOUBLE DENSITY"
19130 UP$(1,1)=CHR$(27):UP$(2,2)=CHR$(
156)
19131 DN$(1,1)=CHR$(27):DN$(2,2)=CHR$(
157)
19132 LT$(1,1)=CHR$(27):LT$(2,2)=CHR$(
158)
19133 RT$(1,1)=CHR$(27):RT$(2,2)=CHR$(
159)
19300 RETURN
```

### The Profile

The main PROFILE routine checks to see if a file called DISKLIB.PRO is available on drive 1. We do this by trying to open the file for reading and immediately closing the file. The STATUS command is then used to check the status of the disk drive. If ERROR=1, everything is OK; the program calls RPROFILE to read the data and then returns to the main program. If ERROR=170, the file was not found on the disk. We call the CPROFILE routine to create the file before returning. For any other error, we print out a short message with the error number and give the user a chance to react (disk drive not on?).

```
11000 REM *****
11001 REM PROFILE: READ/CREATE
11002 REM *****
11010 REM DOES DISKLIB.PRO EXISTS
11020 TRAP 11040
11030 OPEN #1,4,0,"D1:DISKLIB.PRO"
11040 CLOSE #1
11050 STATUS #1,ERROR
11060 IF ERROR=1 THEN GOSUB RPROFILE:R
ETURN
11070 IF ERROR=170 THEN GOSUB CPROFILE
:RETURN
11080 GOSUB SCREEN:POSITION 2,4:"DIS
K ERROR NO. ";ERROR:?:?"CHECK DISK D
RIVE BEFORE CONTINUING."
11090 GOSUB PAUSE:GOTO 11020
```

### SCREEN and PAUSE Routines.

Two short, but useful, routines are introduced here. In 11080, before we print out our message, you see a call to a subroutine called SCREEN. Quite often throughout the program, the screen is cleared before new information is presented. A GRAPHICS 0 command clears the screen. Unfortunately, it also resets the color registers. Since the user selects the screen colors, color registers must be altered after each GRAPHICS 0 command. This is done in the SCREEN routine. Poking a 1 into 752 turns the cursor off.

```
340 REM *****
341 REM SCREEN: PRINT NEW SCREEN
342 REM *****
345 GRAPHICS 0:SETCOLOR 2,BKC,BKL:SETC
OLOR 1,0,CHL:POKE 752,1:RETURN
```

The PAUSE routine gives the user time to absorb information on the screen. It prints out a prompt, centered on the bottom line of the screen, and waits for the user's response before continuing.

```
360 REM *****
361 REM PAUSE: HOLD SCREEN UNTIL KEY
362 REM *****
365 POSITION 7,23:?"(HIT ANY KEY TO C
```



ONTINUE)";GOSUB KEYBD:RETURN

#### Read and Save the Profile

The routines to read and to save the profile data are similar. The TRAP statement assures that the program does not crash if an error is encountered (you run out of data during the read or have a write-protect tab on during the write). Channel #2 is opened for either read (4) or write (8) access to the file D:DISKLIB.PRO. The data read (or written) are the three color variables followed by values for each of the eight elements in the DTYPE array. When we expand the data saved in our profile, we can add the new variables to each routine. The channel is closed when we are finished reading or writing.

```
11300 REM *****
11301 REM RPROFILE: READ PROFILE
11302 REM *****
11310 TRAP 11390
11315 OPEN #2,4,0,"D:DISKLIB.PRO"
11320 INPUT #2,BKC: REM COLOR DATA
11322 INPUT #2,BKL
11324 INPUT #2,CHL
11330 FOR I=1 TO 8: REM DISK DATA
11332 INPUT #2,X
11334 DTYPE(I)=X
11336 NEXT I
11390 CLOSE #2
11395 RETURN
11300 REM
```

```
11400 REM *****
11401 REM SPROFILE: SAVE PROFILE
11402 REM *****
11410 TRAP 11490
11415 OPEN #2,8,0,"D:DISKLIB.PRO"
11420 PRINT #2,BKC: REM COLOR DATA
11422 PRINT #2,BKL
11424 PRINT #2,CHL
11430 FOR I=1 TO 8: REM DISK DATA
11432 X=DTYPE(I)
11434 PRINT #2,X
11436 NEXT I
11490 CLOSE #2
11495 RETURN
```

#### Setting, and Changing, Options

As you've just seen, reading and writing from a data file is a relatively simple process. All that remains to be done now is to get the information we want for the file. Since the steps taken to create the initial information are the same as those needed to change the information, the routines to specify disk, printer, and color settings are kept distinct. When we are initially creating the file (CPROFILE), we print out a message explaining what we are doing and then send the user through each of these change

routines. Later, we use another routine (CHPRO) in the main menu to allow the user to change whatever settings he wants. I'll discuss each of these in turn.

```
11200 REM *****
11201 REM CPROFILE: CREATE PROFILE
11202 REM *****
11205 GOSUB SCREEN: INV=1: ROW=1: B$="DIS
KETTE LIBRARIAN": GOSUB PRcen: INV=0
11207 ROW=3: B$="PROFILE": GOSUB PRcen: P
OSITION 2,6
11210 ? " I need to determine your
standard working environment. I will
store"
11220 ? "this data in a file called
DISKLIB.PRO. The 'PRO' is sh
ort for"
11230 ? "PROFILE. This file will hold
data on the number and kind of dis
k drives,"
11240 ? "the kind of printer, and the
colors you want used with this progr
am.": ?
11250 ? " You can change any of th
ese settings with the CHANGE PROF
ILE option.": ? : ?
11260 GOSUB PAUSE: GOSUB CHDISK: GOSUB C
HPRINT: GOSUB CHCOLOR: GOSUB SPROFILE
11270 RETURN
```

#### Creating the Profile.

You've already been introduced to the SCREEN subroutine. I am sure you understand the print statements as well as the call to the individual routines in 11260. The call, however, to PRcen in lines 11205 and 11207 is new. PRcen prints B\$, centered on the page (hence the name), at the line indicated by ROW. If the value of the INV variable is nonzero, B\$ is printed in inverse video. Thus 11205 prints "DISKETTE LIBRARIAN", centered and in inverse video, on line one and 11207 prints "PROFILE" in the center of line three.

```
370 REM *****
371 REM PRcen: CENTER B$ ON SCREEN
372 REM *****
373 CLEN=LEN(B$): COLUMN=INT((40-CLEN)/
2): POSITION COLUMN,ROW
374 IF INV=0 THEN ? B$: GOTO 376
375 FOR J=1 TO CLEN: ?CHR$(ASC(B$(J,J)
)+128);: NEXT J
376 RETURN
```

The variable CLEN is set to the length of B\$. To center this value on a 40-column page, calculate the spaces left after subtracting CLEN from 40, divide this answer in half, and assign the result to COLUMN. Position the cursor



at COLUMN,ROW. If INV is 0, print B\$ and return. If not, print B\$ in inverse video by adding 128 to the decimal code for every character in B\$. Printing any phrase, in regular text or inverse video, centered on the row of your choice, can be useful in a great many programs. Experiment with PROCEN until you understand how it works.

#### Changing the Profile.

To allow us to update the profile settings, we add one more line to our initial main menu (1045 below). I've also reprinted lines 1030 and 1050 from last month. The menu should display with a box drawn around it. You draw the corners of the box with CTRL Q,E,Z, and C. The top and bottom lines are drawn with CTRL R and the sides with SHIFT =. Once again, since the ATARI graphic symbols don't lend themselves to printing, I've respecified the lines for the top and bottom of the box using the CHR\$ function. Feel free to type the actual graphic characters in your copy if you like.

```
1000 REM *****
1001 REM MENU: PRESENT MAIN MENU
1002 REM *****
1010 GOSUB SCREEN:POSITION 11,1

1030 ? CHR$(17);:FOR I=1 TO 26:? CHR$(
18);:NEXT I:? CHR$(5)

1045 ? "1 8) CHANGE PROFILE      !"

1050 ? CHR$(26);:FOR I=1 TO 26:? CHR$(
18);:NEXT I:? CHR$(3)
```

The CHPRO routine is very similar to the main menu. We present the user with five possible choices. The first three deal with the various options, the fourth saves these changes permanently, and the fifth returns the user to the main menu. The pokes to 82 in 9025 and 9075 move the left margin over to 10 before printing the menu choices and back to 2 when leaving the subroutine. It is used to simplify centering the box.

```
9000 REM *****
9001 REM CHPRO: CHANGE PROFILE
9002 REM *****
9010 GOSUB SCREEN
9020 ROW=1:INV=1:B$="CHANGE PROFILE":G
OSUB PROCEN:INV=0
9025 POKE 82,10:? :?
9030 ? CHR$(17);:FOR I=1 TO 18:? CHR$(
18);:NEXT I:? CHR$(5)
9032 ? "1 1) DISK DRIVE(S) !"
9034 ? "1 2) PRINTER      !"
9036 ? "1 3) COLORS       !"
9038 ? "1 4) SAVE CHANGES !"
9040 ? "1 5) RETURN      !"
```

```
9042 ? CHR$(26);:FOR I=1 TO 18:? CHR$(
18);:NEXT I:? CHR$(3)
9044 ? :? "ENTER CHOICE: ";
9050 GOSUB KEYBD
9060 IF K$<"1" OR K$>"5" THEN GOSUB BAD
KEY:GOTO 9050
9070 ? K$
9072 IF K$="5" THEN GOTO 9100
9075 POKE 82,2
9080 ON VAL(K$) GOSUB CHDISK,CHPRINT,C
HCOLOR,SPROFILE
9090 GOTO 9010
9100 RETURN
```

#### Wrong Answers.

If a key other than the numbers from 1 through 5 is pressed, we go to BADKEY where a short rasping sound warns of an inappropriate choice before checking the keyboard again. You can experiment with the sound command and put in whatever sound you prefer.

```
350 REM *****
351 REM BADKEY: SOUND FOR BAD INPUT
352 REM *****
355 SOUND 0,150,12,10:FOR II=1 TO 20:N
EXT II:SOUND 0,0,0,0:RETURN
```

#### Disk Drive Settings

We want a screen that allows us to easily indicate the number and type of disk drives available. A single keystroke should be sufficient to indicate the status of each drive. There are many different ways to accomplish this. Perhaps you can improve on the method shown here. You might, for example, have the computer check the status of each drive automatically. Be forewarned, however, that it is not a simple task to have the computer change the density of a disk drive from within a BASIC program.

Before examining the BASIC code, let's look at the screen presented to the user:

#### Disk Drive Settings

DRIVE NO.	DRIVE SETTING:
1	single density
2	DOUBLE DENSITY
3	NOT AVAILABLE
4	NOT AVAILABLE
5	NOT AVAILABLE
6	NOT AVAILABLE
7	NOT AVAILABLE
8	NOT AVAILABLE



Use (U) and (D) to change options.  
 Use (L) and (R) to change settings.  
 Use RETURN when finished.

Initially, the setting for drive 1 is in inverse video to indicate the drive we are dealing with. The arrow keys (shown by the U, D, L, and R in the prompt above only because of the difficulty in printing these symbols) are used to make selections. The up and down arrows allow the user to select a disk drive. The left and right arrows change the setting for that drive among the different possibilities.

The prompt at the bottom of the screen is generated by a routine called HPROFILE (for HELP PROFILE). It is also used to assist in the selection of the various color options.

```
11600 REM *****
11601 REM HPROFILE: HELP FOR PROFILE
11602 REM *****
11610 POSITION 2,19: ? "-----"
11620 POSITION 2,20: ? "Use ";UP$;" and
      ";DN$;" to change options."
11630 POSITION 2,21: ? "Use ";LT$;" and
      ";RT$;" to change settings."
11640 POSITION 2,22: ? "Use RETURN when
      finished."
11650 RETURN
```

The CHDISK routine has two basic parts. It displays the initial screen shown above (13030 to 13110). Then, it goes to the keyboard for the user's response and changes the screen accordingly until the RETURN key is pressed (13120 to 13270).

```
13000 REM *****
13010 REM CHDISK: DRIVE SETTINGS
13020 REM *****
13030 GOSUB SCREEN:ROW=1:INV=1:B$="DIS
K DRIVE SETTINGS":GOSUB PROCEN:INV=0
13040 REM Print Initial Screen
13050 POSITION 7,4: ? "DRIVE NO.":POSIT
ION 19,4: ? "DRIVE SETTING:"
13060 POSITION 7,5: ? "-----"
13070 FOR I=1 TO 8
13080 ROW=I+6:I1=14*(DTYPE(I))+1:I2=I1
+13
13090 POSITION 10,ROW: ? I:GOSUB 13270
13100 NEXT I
13110 GOSUB HPROFILE
13120 INDEX=1
13125 REM Change Settings as Required
```

```
13130 IF INDEX=9 THEN INDEX=1
13140 IF INDEX=0 THEN INDEX=8
13150 ROW=INDEX+6:DTY=DTYPE(INDEX):I1=
14*DTY+1:I2=I1+13
13160 POSITION 19,ROW:FOR J=I1 TO I2: ?
CHR$(ASC(DSTAT$(J,J))+128):NEXT J
13170 GOSUB KEYBD
13180 IF K=155 THEN GOSUB 13270:RETURN
13190 IF K$="-" OR K$="_" OR K$=CHR$(2
8) THEN INDEX=INDEX-1:GOSUB 13270:GOTO
13130
13200 IF K$="=" OR K$="!" OR K$=CHR$(2
9) THEN INDEX=INDEX+1:GOSUB 13270:GOTO
13130
13210 IF K$="+" OR K$="\ " OR K$=CHR$(3
0) THEN DTY=DTY-1:GOTO 13240
13220 IF K$="*" OR K$="^" OR K$=CHR$(3
1) THEN DTY=DTY+1:GOTO 13240
13230 GOSUB BADKEY:GOTO 13170
13240 IF DTY=3 THEN DTY=0
13250 IF DTY<0 THEN DTY=2
13260 DTYPE(INDEX)=DTY:GOTO 13150
13270 POSITION 19,ROW: ? DSTAT$(I1,I2):
RETURN
```

There are some tricks used in achieving the effects we want. These will be familiar to experienced programmers, but if you are just starting, all of the code may not be intuitively obvious. I want to take some time to explain how to print part of a string variable since this is a technique used quite often.

A new variable is found in a one-line subroutine at 13270. DSTAT\$ holds 42 characters arranged as 3 groups of 14 characters each (see lines 19020 and 19120 in INIT above). The 14-character expressions represent the three possible states of a disk drive (NOT AVAILABLE, SINGLE DENSITY, and DOUBLE DENSITY) and correspond to the numbers in DTYPE (0, 1, and 2). If the value of DTYPE(n) is 0 for drive n, we want to print DSTAT\$(1,14), the first 14 characters of DSTAT\$. Similarly, if DTYPE(n) is 1, then we print the second 14 characters, DSTAT\$(15,28), and if DTYPE(n) is 2, we print the last 14 characters, DSTAT\$(29,42). It is no accident that these expressions are all set to an identical length (note the extra space between NOT and AVAILABLE in line 19120). A mathematical formula is used to calculate the first (I1) and last (I2) character we want printed from DSTAT\$:

$$I1 = 14 * DTYPE(n) + 1$$

$$I2 = I1 + 13$$

The value of DTYPE(n) is either 0, 1, or 2. For each of these values, calculate I1 and I2 and convince yourself that this formula works. Do you understand why? The formula is used in 13080 to determine the beginning and ending



characters needed from DSTAT\$ before calling 13270 to print them.

The variable INDEX keeps track of the disk drive. It is initially set to 1. When the up and down arrows are pressed, the value of INDEX decreases or increases accordingly. Lines 13130 and 13140 keep us within the 1 to 8 limits. In 13150 we calculate the appropriate row, the value in DTYPE for that drive (the variable DTY), and the starting and ending positions needed from DSTAT\$. In 13160 we print the appropriate expression from DSTAT\$ in inverse video and then go to the keyboard for the user's instructions.

If the RETURN key (K=155) is pressed, we are done. If not, we check to see if the up or down arrows were pressed. (We check all three possibilities so that pressing the 'arrow', SHIFT 'arrow', or CTRL 'arrow' all generate the appropriate response. If up or down is pressed, we change the value of INDEX, print DSTAT\$ in normal letters, and then jump to 13130 where INDEX is checked and the new selected value is printed in inverse video.

Lines 13200 and 13210 check for the left or right arrows. If either of these are pressed, the value of DTY is changed (staying within the 0-2 bounds), the corresponding value in DTYPE is changed, and the new DSTAT\$ setting is printed.

If the key wasn't an arrow key or the RETURN key, we go to BADKEY before checking the keyboard again.

#### Printer Settings.

We are not yet ready to implement the printer profile but we can include a dummy routine that will be written later:

```
14000 REM *****
14010 REM CHPRINT: CHANGE PRINTER SETTINGS
14020 REM *****
14030 GOSUB SCREEN
14040 ? "PRINTER SETTINGS":GOSUB PAUSE
14050 RETURN
```

```
50550
6001150500
```

#### Setting Colors.

Setting our own screen colors is fun, but it does introduce some programming complexity. Once again, let's start by looking at the initial screen display:

#### COLOR SETTINGS

#### BACKGROUND COLOR

#### Turquoise Blue

#### BACKGROUND LUMINANCE

<--Darker----->

#### CHARACTER LUMINANCE

<-----Med Light----->

Use (U) and (D) to change options.  
Use (L) and (R) to change settings.  
Use RETURN when finished.

The technique we use is similar to that used to specify disk drive settings but there are some important differences here. Initially, the text describing the background color, "Turquoise Blue", is shown in inverse video to indicate that is the setting we are adjusting. If the left or right arrow key is pressed, the background color on the screen changes as does the phrase describing the color. Pressing the up or down arrow keys selects among BACKGROUND COLOR, BACKGROUND LUMINANCE, AND CHARACTER LUMINANCE. On either of the LUMINANCE settings, changing the luminance is reflected both by the color change as well as a change in the descriptive text. When the RETURN key is pressed, the chosen colors remain in effect throughout the program. (If you are typing this code in, be sure to replace all of the GRAPHICS 0 statements last month with GOSUB SCREEN.)

Although the technique is similar to that used in the disk drive case, the implementation is trickier. We don't have an INDEX that allows us to move from one option to the other and at the same time tells us our position on the screen and where to look for the option setting. What's more, the settings available for background color are entirely different from those used to indicate background and character luminance.

We could dimension a very long string variable to hold all the color and luminance settings. But that would use up a lot of memory for little benefit. Instead, we will use READ and DATA statements to get the information we need and put it in B\$. Two subroutines are included to get the text describing the BKC, BKL, and CHL settings. The first, RDCOL, restores the data pointer to the line containing the appropriate color description, reads B\$, and returns. As you can see, the order of the DATA statements is important.

```
15200 REM *****
15201 REM RDCOL: READ COLOR DATA
15202 REM *****
15210 RESTORE 15220+BKC:READ B$:RETURN
15220 DATA Black to White
15221 DATA Brown to Gold
15222 DATA Orange to Yellow
15223 DATA Terra cotta to Pink
```



```

15224 DATA Mulberry to Magenta
15225 DATA Violet to Lavender
15226 DATA Indigo to White
15227 DATA Sky Blue
15228 DATA Royal Blue to Baby Blue
15229 DATA Turquoise Blue
15230 DATA Ultramarine to Powder Blue
15231 DATA Midnight Blue to Aquamarine
15232 DATA Sea Green to Turquoise Green
15233 DATA Forest Green to Kelly Green
15234 DATA Olive
15235 DATA Khaki to Yellow

```

The RDLUM subroutine is similar. Since both BKL and CHL need a luminance setting, the RDLUM has been generalized to handle both cases. The luminance value (either BKL or CHL) is placed in a variable named LVAL before this routine is called. The luminance values can be 0,2,4,6,8,10,12, or 14. Line 15310 converts these numbers to 1,2,3,4,5,6,7, or 8 and places the result in L. Line 15320 restores the data pointer to the appropriate text, reads in B\$, and returns.

```

15300 REM *****
15301 REM RDLUM: READ LUMINANCE
15302 REM *****
15310 L=INT((LVAL+2)/2)
15315 RESTORE 15320:L:READ B$:RETURN
15321 DATA <Darkest----->
15322 DATA <--Darker----->
15323 DATA <----Dark----->
15324 DATA <-----Med Dark----->
15325 DATA <-----Med Light----->
15326 DATA <-----Light----->
15327 DATA <-----Lighter--->
15328 DATA <-----Lightest>

```

Now that we know how to get text describing the color and luminance settings, we can print out the initial screen:

```

15000 REM *****
15010 REM CHCOLOR: COLOR SETTINGS
15020 REM *****
15030 GOSUB SCREEN
15040 INV=1:ROW=1:B$="COLOR SETTINGS":
GOSUB PRCEM
15050 INV=0:ROW=4:B$="BACKGROUND COLOR":
GOSUB PRCEM
15060 ROW=8:B$="BACKGROUND LUMINANCE":
GOSUB PRCEM
15070 ROW=12:B$="CHARACTER LUMINANCE":
GOSUB PRCEM
15080 GOSUB RDCOL:ROW=5:GOSUB PRCEM
15090 LVAL=BKL:GOSUB RDLUM:ROW=9:GOSUB
PRCEM
15100 LVAL=CHL:GOSUB RDLUM:ROW=13:GOSUB
B PRCEM

```

```

15110 GOSUB HPROFILE
15120 GOTO 15400

```

By this time, you should recognize everything going on in setting up the display screen. Our PRCEM subroutine comes in handy here where every line is centered. After printing the help prompt in 15110, the initial screen is done and the program jumps to 15400 where we handle the user interaction.

The remainder of the CHCOLOR routine is divided into four distinct parts, one each for the color options and one to read and interpret keyboard responses. Changes in background color are handled at 15400, changes in background luminance at 15500, and changes in character luminance at 15600. The keyboard is queried at 15700. We want to use the arrow keys again to indicate selections. The up and down arrows should move us among the three possible options while the left and right arrows change the settings for an option. The RETURN key indicates that we are satisfied with our choices.

Reading the Keyboard.

Let's look first at the code that checks which key is pressed:

```

15700 GOSUB KEYBD:IF K=155 THEN RETURN
15710 IF K$="-" OR K$="!" OR K$=CHR$(2
8) THEN INV=0:GOSUB PRCEM:GOTO AFTER
15720 IF K$="-" OR K$=" " OR K$=CHR$(2
9) THEN INV=0:GOSUB PRCEM:GOTO BEFORE
15730 IF K$="+" OR K$="\ " OR K$=CHR$(3
0) THEN CHANGE=-1:GOTO HERE
15740 IF K$="*" OR K$="^" OR K$=CHR$(3
1) THEN CHANGE=+1:GOTO HERE
15750 GOSUB BADKEY:GOTO 15700

```

If the RETURN key is pressed, we are all done so we exit the CHCOLOR subroutine. Lines 15710 and 15720 check for up or down arrows. If one of these is pressed, we reverse the inverse flag (INV=0), print B\$ (which will hold our color or luminance data), and then go to the option either BEFORE or AFTER the current one. The next two lines check for left or right arrows. If pressed, we want to change the current option setting. We do so by setting the value of a variable called CHANGE to either +1 or -1 and then going to the current option code represented by the variable HERE. If any other key is pressed, we call BADKEY and then check the keyboard again.

Now let's look at the code to change background colors:

```

15400 REM SELECT BACKGROUND COLORS
15410 ROW=5:AFTER=15500:HERE=15440:BE
FORE=15600
15420 INV=1:GOSUB RDCOL:GOSUB PRCEM

```



```

15500 REM SELECT BACKGROUND LUM
15510 ROW=9:AFTER=15600:HERE=15540:BEF
ORE=15400
15520 INV=1:LVAL=BKL:GOSUB RDLUM:GOSUB
PRCEN
15530 GOTO 15700
15540 REM CHANGE BACKGROUND LUM
15550 BKL=BKL+2*CHANGE
15552 IF BKL>15 THEN BKL=0
15554 IF BKL<0 THEN BKL=15
15560 INV=0:B*=BLANK$:GOSUB PRCEN
15570 SETCOLOR 2,BKC,BKL:GOTO 15520

```

```

15600 REM SELECT CHARACTER LUM
15610 ROW=13:AFTER=15400:HERE=15640:BE
FORE=15500
15620 INV=1:LVAL=CHL:GOSUB RDLUM:GOSUB
PRCEN
15630 GOTO 15700
15640 REM CHANGE CHARACTER LUM
15650 CHL=CHL+2*CHANGE
15652 IF CHL>15 THEN CHL=0
15654 IF CHL<0 THEN CHL=15
15660 INV=0:B*=BLANK$:GOSUB PRCEN
15670 SETCOLOR 1,0,CHL:GOTO 15620

```

Well, that does it for this month. Next month we will, finally, read those disk directories.

## THE SAGA OF PLANDEFRON SEVEN THE FOX FIGHTS ALONE

Planetary Defense Squadron Seven had just proven itself as a fighting unit. Only two days before they had driven off an attack upon Collins Base and its vital stores, but at a terrible cost. Now with only one of the eight ships... Foxtrot, "the Fox"...able to fight, the entire system lay in grave peril...

for the sentinel posts had just reported an incoming craft, its zigzag maneuvers a certain prelude to attack. Then the outposts fell silent.

\*\*\*\*\*

SEVEN FOX is a game for one to five players of varying skill levels who each take a separate action station as the crew of the FOX in its desperate bid to intercept this attacker bent on destroying an entire planet. The game requires an ATARI\* computer with at least 16K of memory, (24K with 810 Disk) a model 410 Program Recorder, a BASIC language cartridge, a pair of knob ("paddle") controllers, and a pair of joysticks.

Special AUDIO TRACK FEATURES: a "radio play" to provide background, and an audio-visual "briefing" for beginners, available on the cassette format make this game a unique tool for teaching three-dimensional rectangular coordinates. The team-play approach also provides practice in cooperation and leadership.

```

15430 GOTO 15700
15440 REM CHANGE BACKGROUND COLOR
15450 BKC=BKC+CHANGE
15452 IF BKC>15 THEN BKC=0
15454 IF BKC<0 THEN BKC=15
15460 INV=0:B*=BLANK$:GOSUB PRCEN
15470 SETCOLOR 2,BKC,BKL:GOTO 15420

```

In 15410 we indicate the appropriate ROW value for the text description of the color values. We also set values for three variables. AFTER indicates where the code for the next option begins, HERE indicates where to go if the setting of this option is changed, and BEFORE indicates where to go if the previous option is desired. We set the inverse flag (INV=1), read in the appropriate color, print it and then go check the keyboard. If a change in this setting is desired, the program will return to HERE which is line 15440. The variable BKC is then changed by + or - 1. Lines 15452 and 15454 keep us within the 0 to 15 limits. Line 15460 clears the current text for color description and the next line changes the appropriate color register before jumping to the beginning of this option's code where we get the new color text and print it.

If you can follow the logic above, you'll see that the remaining two options are handled in exactly the same way:



Cassette or  
5 1/4 inch Floppy Disk (no briefing)

\$19.00 each

Special terms for qualifying school systems.

Contact me at DCAUG meetings or write to:

Bennett Rutledge, CDP  
327 South Wayne Street  
Arlington, Va. 22204-2126

\*ATARI is a registered trademark of Atari, Inc.



## DISK DRIVE INTERFACE (via SIO)

Contributed by Bruce Blake

This article contains information and data from the MYDOS 3.05 Manual, the ATARI Technical Reference Notes, and the ARCHIVER/EDITOR User's Manual.

The physical disk drives and diskettes are external to the ATARI home computers and are normally attached to the "serial interface connector" on the right side of the computer. The software in the operating system (OS ROMs) that access the devices attached to the serial interface connector is called the "serial I/O driver" or SIO for short. The operating system uses this driver to pass all commands and information to and from the disk drive and other peripherals. Several commands were defined by ATARI to communicate with the 810 disk drive, but an extended set of commands is required to support double density functions.

The minimum set of disk drive functions are:

Device	Unit	Command	Direction	Byte Ct.	Aux. Bytes	Function
\$31	Drive#	\$21 !	From Drive	128/256	1 to 720	FORMAT DISK
\$31	Drive#	\$50 P	To Drive	128/256	1 to 720	WRITE (no verify)
\$31	Drive#	\$52 R	From Drive	128/256	1 to 720	READ
\$31	Drive#	\$53 S	From Drive	4	1 to 720	READ STATUS
\$31	Drive#	\$57 W	To Drive	128/256	1 to 720	WRITE (verify)

Except for the STATUS and CONFIGURE commands, the byte count is always 128 for a small sector drive, and is 128 for the first three sectors (1, 2, and 3) of a large sector drive. All other sectors on a large sector drive are 256 bytes long.

The first byte returned by the READ STATUS command is expected to indicate the sector size. If bit 5 is a 1 then the sectors are large (256 bytes), otherwise, they are small (128 bytes). Bit 3 of this byte will indicate if the disk is write protected or not. If bit 3 is a 1 then the disk is write protected. It seems that bit 4 will indicate if the disk drive is configurable or not. If bit 4 is set the drive is configurable as single or double density.

Status Request Data	First Byte
\$00,\$FF,\$E0,\$00	SD WP Off ATARI 810 Only
\$08,\$FF,\$E0,\$00	SD WP On ATARI 810 Only
\$10,\$FF,\$E0,\$00	SD WP Off
\$30,\$FF,\$E0,\$00	DD WP Off
\$18,\$FF,\$E0,\$00	SD WP On
\$38,\$FF,\$E0,\$00	DD WP On

Byte two of the floppy disk Controller status byte. A low bit indicates the error exists. .

Bit	Read	Write	Notes
7	Not Ready	Not Ready	Always Clear
6	Data Error #1	Write Protect	
5	Data Error #2	Write Fault	
4	Record Not Found	Record Not Found	Sector Missing
3	CRC Error	CRC Error	
2	Lost Data	Lost Data	Should Not Happen
1	DRQ	DRQ	Always Clear
0	Busy	Busy	Always Clear

Byte three is the number of seconds required to format a disk.

Byte four is not used.

The new commands added are:

Device	Unit	Command	Direction	Byte Ct.	Aux. Bytes	Function
\$31	Drive#	\$4E N	From Drive	12	1 to 720	READ Configuration
\$31	Drive#	\$4F O	To Drive	12	1 to 720	WRITE Configuration



These commands permit reconfiguration of a disk drive on demand. These commands also support identification of a disk drive as single or double density, 5 or 8 inch and one or two sided. The individual bytes are defined as follows:

byte 0: Tracks per side (40 for a standard disk drive)

byte 1: Disk Drive Step Rate (as defined by Western Digital)

Code value	8 inch rate	5 1/4 inch rate
0	3 ms/track	6 ms/track
1	6 ms/track	12 ms/track
2	10 ms/track	20 ms/track
3	15 ms/track	30 ms/track

byte 2: Zero (high byte of sectors/track)

byte 3: Sectors/Track (18 for standard diskettes)

byte 4: Side Code (0=single sided, 1=double sided)

```
byte 5:  Disk Type Code --
```

bit 2: 0=single density, 1=double density

bit 1: 0=5 1/4 inch diskette, 1=8 inch diskette drive

byte 6: High byte of Bytes/Sector (0 for ATARI 810 compatible)

byte 7: Low byte of Bytes/Sector (128 for ATARI 810)

```
byte 8: Translation control -- (Not valid on TRAK AT-D2)
```

bit 7: 1=40 trk. disk I/O on an 80 trk. drive

bit 6: Always 1 (to indicate drive present)

bit 1: 1=Handle sectors 1, 2, and 3 as full size sectors

bit 0: 1=Sectors number 0-17 (for example) not 1-18

bytes 9-11 are not used (and should be zero)

Send Configure Command Data

SD 40,0,0,18,0,0,\$0,\$80,1,0,0,0

DD 40,0,0,18,0,4,\$1,\$00,1,0,0,0

### Recieve Drive Configuration Data

SD 40,3,0,18,0,0,\$0,\$80,\$FF,\$FF,\$FF,\$FF

```
DD 40,3,0,18,0,4,$1,$00,$FF,$FF,$FF,$FF
```

An additional requirement is that the disk drive automatically switch density if required to read sector 1. This is necessary if one is to 'BOOT' either single or double density disks.

The following program will let you interrogate and configure your disk drives

```

10 REM <<<<<<<<<<>>>>>>>>>>>>>>
20 REM <<< PERCOM DISK CONTROL >>>
30 REM <<< SAMPLE PROGRAM >>>
40 REM <<< >>>
50 REM <<< UPDATED BY >>>
60 REM <<< BRUCE BLAKE >>>
70 REM <<<<<<<<<<>>>>>>>>>>>>>>
100 DIM BUF$(12):REM OPTION TABLE
110 DIM R$(1):REM RESPONSES GO HERE
120 Z=ADR("1234")

```

```

130 DATA 104,76,89,228
140 RESTORE 130:FOR I=0 TO 3
150 READ X:POKE Z+I,X:NEXT I
160 DCB=768:DRIVE=1
170 DATA 49,1,78,64,0,0,1,0,12,0,1,0
180 REM DEVICE,UNIT,COMMAND,STATUS(DIR),
190 REM BUFFER ADDR,TIMEOUT,SECTOR ADDR
200 RESTORE 170:FOR I=0 TO 11:REM BUILD THE DCB
210 READ X:POKE DCB+I,X:NEXT I
220 A=ADR(BUF$):REM INSERT BUFFER ADDRESS
230 X=INT(A/256):POKE DCB+5,X:POKE DCB+4,A-X*256
240 TRAP 250:PRINT CHR$(125):PRINT "Drive
Number";:INPUT DRIVE
250 TRAP 45678:POKE DCB+1,DRIVE:REM SET DRIVE IN DCB
260 X=USR(Z):REM READ THE OPTION TABLE
270 IF PEEK(DCB+3)<>1 THEN PRINT "Drive NOT
Configurable":GOSUB 590:END
280 PRINT "      Number of Tracks=";PEEK(A)
290 PRINT "      Step Rate Code=";PEEK(A+1)
300 PRINT "      Sectors/Track=";PEEK(A+3)
310 PRINT "      ";PEEK(A+4)+1;" Sided Drive"
320 IF PEEK(A+5)=0 THEN PRINT "      Single
Density":GOTO 340
330 IF PEEK(A+5)=4 THEN PRINT "      Double Density"
340 PRINT "      ";PEEK(A+6)*256+PEEK(A+7);"
Bytes/Sector"
360 PRINT :PRINT "Make it Single or Double density ";
370 INPUT R$
380 DATA 49,1,79,128,0,0,1,0,12,0,1,0
390 RESTORE 380:FOR I=0 TO 11:REM BUILD THE DCB
400 READ X:POKE DCB+I,X:NEXT I
410 IF R$="D" THEN 460
420 POKE A+5,0:"Make Drive Single Density":PRINT
430 POKE A+6,0
440 POKE A+7,128
450 GOTO 490
460 POKE A+5,4:"Make Drive Double Density":PRINT
470 POKE A+6,1
480 POKE A+7,0
490 A=ADR(BUF$):REM INSERT BUFFER ADDRESS
500 X=INT(A/256):POKE DCB+5,X:POKE DCB+4,A-X*256
510 POKE DCB+1,DRIVE:REM SET DRIVE IN DCB
520 X=USR(Z):REM UPDATE DRIVE OPTIONS
530 IF PEEK(DCB+3)<>1 THEN PRINT "Drive Update
Error":GOTO 550
540 GOSUB 590
550 DATA 104,76,224,7
560 RESTORE 550:FOR I=0 TO 3
570 READ X:POKE Z+I,X:NEXT I
580 X=USR(Z):END :REM UPDATE DOS
590 DATA 49,1,83,64,0,0,1,0,4,0,1,0
600 REM DEVICE,UNIT,COMMAND,STATUS(DIR),
610 REM BUFFER ADDR,TIMEOUT,SECTOR ADDR
620 RESTORE 590:FOR I=0 TO 11:REM BUILD THE DCB
630 READ X:POKE DCB+I,X:NEXT I
640 A=ADR(BUF$):REM INSERT BUFFER ADDRESS

```



```

130 DATA 104,76,89,228
140 RESTORE 130:FOR I=0 TO 3
150 READ X:POKE Z+I,X:NEXT I
160 DCB=768:DRIVE=1
170 DATA 49,1,78,64,0,0,1,0,12,0,1,0
180 REM DEVICE,UNIT,COMMAND,STATUS(DIR),
190 REM BUFFER ADDR,TIMEOUT,SECTOR ADDR
200 RESTORE 170:FOR I=0 TO 11:REM BUILD THE DCB
210 READ X:POKE DCB+I,X:NEXT I
220 A=ADR(BUF):REM INSERT BUFFER ADDRESS
230 X=INT(A/256):POKE DCB+5,X:POKE DCB+4,A-X*256
240 TRAP 250:PRINT CHR$(125):PRINT "Drive Number";:INPUT
DRIVE
250 TRAP 45678:POKE DCB+1,DRIVE:REM SET DRIVE IN DCB
260 X=USR(Z):REM READ THE OPTION TABLE
270 IF PEEK(DCB+3)<>1 THEN PRINT "Drive NOT
Configurable":GOSUB 590:END
280 PRINT "    Number of Tracks=";PEEK(A)

```

## Atari Scuttlebits

by Bob Kelly

Last month I indicated that the primary subject of discussion for this month's column would be Commodore Electronics and their marketing strategy. While this subject deserves attention - particularly since their strategy appears to be "caveat emptor", it will be deferred in order to discuss the introduction of the IBM PCjr.

IBM intends to spend approximately \$40 million in 1984 on advertising the jr. Shipment of the jr. to dealers is expected by mid-January. It is worth focusing this column on the market strategy and pricing policy of IBM in order to permit an initial comparative judgement on which machine, the Atari XL or the jr., is the more versatile computer for home applications considering technical and cost factors.

As is known, IBM in November announced the introduction of its long-awaited PCjr. Competing firms reacted prior to the announcement in a number of ways, most of them defensively (e.g. postponing the introduction of new products - give credit here to Apple for its aggressive efforts). It has been over two months since the announcement of the "jr." and a clear picture of the machine and its capabilities is only now beginning to emerge.

Initially, let's look at the PCjr.'s pricing structure and compare it to the Atari 800XL. While numbers may be boring to some, they are revealing in this case since they implicitly signal what general income level IBM seeks to market its product to. Take a minute to carefully examine the table below:

	PCjr.	800XL
-Computer (64K)*	\$670	\$250
-Disk Drive	\$480 (360K)	\$420 (Trak)
-Printer (Dot Matrix)	\$300	\$300
-RF Modulator for TV	\$ 30	---
-DOS (Assume purchase of Dr.)	\$ 65 (2.1)	Standard
-300 Baud Modem	\$200 (card)	\$200**
-Parallel Printer Port	\$100***	[Trak Dr.]
-80 Column Capability	Yes	Yes
-Enhanced GR. & sound	\$75 (MS cartridge)	Standard
-Player/Missile Graphics	No	Yes
-Monitor	\$300	\$300
Total	\$2,220.00	\$1,470.00

\* The jr. includes 64K RAM plus 64K ROM; Atari ROM is smaller by a considerable amount (8K). Both include built-in Basic and a standard 40-column video display.

\*\* Cost of interface to connect modem to joystick port plus modem itself.

\*\*\* Jr. requires an adapter for parallel printer port; printer port is standard on Trak Drive shown here.

# ROBINS Inc.

## THE SMALL COMPUTER SUPPLIES STORE

6 OUTLET SURGE AND  
SPIKE SUPPRESSOR  
PRINT WHEELS

SHOP BY PHONE

WE SHIP BY UPS

PRICES SLASHED ON RIBBONS

FOR THE FOLLOWING PRINTERS

[B=brand C=compatible]

EPSON FX/MX/RX 80[C] 3.95  
EPSON MX/FX 100[C] 9.95  
PROWRITER/NEC 8023[C] 3.95  
DIABLO HYTYPE 2 FILM[C] 3.50  
DIABLO HYTYPE 2 FILM[B] 4.25  
QUME FILM[C] 1.90  
QUME FILM[B] 2.80  
QUME IV FILM[C] 3.45  
NEC 5500/7700 FILM[C] 3.30  
NEC 3500/2000 film[B] 11.90

### RELOAD YOUR OWN!

Diablo, Qume, NEC film 1.50  
Epson 80, NEC 8023, Itoh 1.95  
Epson 100 3.95

### PAPER

CLEAN EDGE 18 and 20 #26.95

LABELS 3 1/2x15/16 5k 16.00

DIABLO Plastic \$6.40  
Metal \$41.50/49.95  
620 \$11.40  
QUME \$6.40/7.90  
SMITH-CORONA TP-1 \$4.90  
OLIVETTI \$15.50  
OLYMPIA EC \$26.50  
BROTHER \$24.95  
DAISYWHEEL II \$27.50

HEAD CLEANING KIT 15.00

### FLOPPY DISKS

5 1/4 ss/dd

SENTINAL 22.50  
Kybe, 3M, Verbatim 26.50

BROWN ds/dd 29.95  
BROWN ds/(96 tpi) 39.95

## 560-5900

8304 Hilltop Road  
Fairfax, Virginia 22031

(Merrifield Area)

Hours: Monday thru Friday 9-5, Saturday 10-2



The comparison yields a \$750 difference in total cost in favor of the Atari 800XL. Simply put, a PCjr. would cost almost 55 percent more than the Atari system configured in the table. The question is what do you get for this premium that you don't have in the Atari?

Perhaps the most important consideration for many consumers is an institutional one - the IBM name. On a technical basis, the PCjr.'s RAM memory can be expanded upward (press accounts indicate that it can exceed 600K). On the other hand, according to industry reports, the upper limit for the 800XL series RAM expansion being planned by 3rd party dealers is 256K. The jr. has Basic Language capabilities that cannot be duplicated by the Atari unless 3rd party software programs are added to RAM Memory. In this particular comparison, think of building-in the excellent MMG software program "Basic Commander & Debugger" to your Atari Basic and you will have an idea of the power of the PCjr.'s Basic. This is clearly a plus for the PCjr.

The PCjr. also has a remote keyboard; the Atari does not. The drawbacks to the remote keyboard are: (1) If another jr. is operated nearby it must be run from a direct connect cord (\$20) and (2) the operator must be within 20 feet of the system unit. On balance, I see the remote keyboard as a potential marketing advantage for the jr. and a technical headache for the User and IBM service stations. Both jr. and the 800XL have self diagnostic programs built in. Both offer two joystick ports and light pen capability - no difference here.

The IBM has output jacks for both composite video and RGB(Hi-Res) direct drive video. Atari offers only composite video capability. This RGB capability of the PCjr. sounds exciting but remember, you must pay extra for the MicroSoft cartridge (\$75). With this cartridge, the User can obtain expanded sound and access 4 colors in the 320 x 200 pixels mode, 2 colors in the 640 x 200 mode, and 16 colors in the 160 x 200 mode (not to mention the extra cost for a RGB monitor). The expanded Basic Cartridge requires an additional 32K of ROM for a total of 64K resident in ROM. The 64K version of the PCjr. has about 45K for Basic programming - not much different from the 800XL.

While it would be interesting to personally speculate on the graphics capabilities of the PCjr., it is probably best to quote Ken Williams, President of Sierra-on-Line, "Consumers who are looking to purchase a home computer solely for game playing would do better to buy an Atari or Commodore 64... There is no way our game, Frogger, will ever look as good on the PCjr. as it does on the Atari or Commodore 64" (InfoWorld - Dec. 1983). Given that the jr. does not have player/missile capabilities, but rather relies on page flipping - buy Atari if you want graphics for programming or game purposes.

In addition to the price advantage and graphics capabilities, the Atari offers a keyboard superior to the PCjr. "chiclet keys". This is not insignificant in relation to the expanded attention being given to word processing

applications for home computers.

In sum, I would say that Atari has a significant marketing advantage at the low and middle range of the home computer market. The very high end has to be, at present, an advantage for IBM for two reasons: (1) The relationship to business applications and (2) the built-in advanced Basic programming capabilities. This view is strengthened by the fact that the jr., as noted in our comparison, comes with a DS/DD disk drive and greater potential RAM expansion than the Atari XL (600K+ versus 256K). Although, from a practical standpoint the 62-key keyboard is a drawback as well as the flimsy construction of the "chiclet" keys themselves for business applications.

Currently, many software houses are scrambling to develop programs for the PCjr. in order to capitalize on IBM's name recognition and to expand their market share. Some firms have stockpiled large inventories in anticipation of jr.'s introduction. This strategy is not without some risk since IBM plans to use only its outlets for sales - and service - no independent dealers. Further, as can be seen above, IBM's pricing strategy is slanted toward the higher end of the home computer market. Thus, with relatively limited sales outlets and a higher price structure, sales may not grow as fast as anticipated by some software houses - particularly for those relying solely on PCjr. game sales. In my opinion, 6 to 8 months from now, a few software houses may find themselves in deep water.

Will Atari take advantage of its competitive position in the middle to low end and/or will it attempt to challenge in the high end? Your guess is as good as mine. Someone in Atari's strategic analysis or marketing department must have put together a table similar to the one shown here and hopefully it should contain more in-depth comparisons to aid in the development of marketing strategies. I say "hopefully" since the quote on the National Archives Building comes immediately to mind when speaking of Atari and marketing in the same sentence - "What Is Past Is Prologue". I certainly hope not!

### UPCOMING EVENTS

January	7-10	Consumer Electronics Show	Las Vegas
	19-21	Data West	Pasadena, CA
	30-		
February	2	Communication Networks '84	Washington, DC
	3-5	National Software Show	Miami Beach
	7-9	Cadcon West	San Francisco
	14-16	Computer Science Conf.	Philadelphia
	20-22	1984 Office Automation Conf	Los Angeles
	21-23	Softcon	New Orleans



## NIBBLES AND BITS

by Jay Gerber

Hello fellow Atari enthusiasts and welcome to Nibbles and Bits. Each month I plan to write two reviews of current software for the Atari computers. The categories should cover the entire spectrum, from education to business.

This month I will review two new arcade-type games: Preppie II; and Lode Runner.

Preppie II, from Adventure International, is a whimsical game in which you are the "up and coming prepster" trying to leave a fiendish network of mazes alive. Armed with a paint bucket that contains blue and gold pinstripe paint, and a special cloak that renders the wearer invisible, you must escape fifteen different mazes on five levels of play. Accompanied by a turn-of-the-century medley of tunes, you, Wadsworth Overcash, find yourself in a formidable maze with revolving doors and huge frogs who can't wait to stomp you out of existence. After you finally cover the entire floor of the , you walk out, only to discover yourself on a busy thoroughfare, filled with vicious golf carts and grass mowers. If you can paint the entire highway, you must face another frog maze.

Gameplay on Preppie II is as smooth and fun as was Preppie I, the original. Music is every bit as good, as are the stunning graphics. There is even a cute cartoon at the end of every level. Despite all this, the game gets boring after only a short while. Unlike Preppie where each new level contained a new challenge, Preppie II, has slightly altered mazes, and smarter frogs as you progress, but this is not enough to keep, at least, my interest, anyway. On a scale of one to ten, Preppie II rates a '5'.

PREPPIE II from Adventure International  
32K disk (\$34.95)  
cassette (\$29.95)

Lode Runner, from Broderbund, is an action/adventure game in which "you are a Galactic Commando deep in enemy territory." Your mission is to retrieve stolen gold from the Bungeling Empire. In your way are trapdoors, cunningly evil enemy soldiers, and many puzzles in which you must use your wits, rather than your reflexes to solve. In the 150 different rooms you will have to reaverse canyons using hand-over-hand bars and ladders, digging holes which the enemy soldiers fall into. After entering a room, you must find all gold chests on that board. This isn't easy, especially when three or four guards are chasing you. When you have retrieved all the gold on a level, a ladder appears, and you climb up to the next level.

Lode Runner seems to be a mixture of several top selling games. First is the scenario of a hero running from board to board in search of treasure like Epyx's Jumpman and Jumpman Junior (soon to be reviewed); the unique digging to kill feature of Apple Panic (also by Broderbund); and Donkey

Kong's "up the ladder to the next board." The element that makes Lode Runner far superior to these games is it's board generator. With this tool, you can create your own boards to play at any time. Ladders, floor sections, gold chests, trapdoors, hand-over-hand bars, even enemies can be located anywhere on the playing field. It is this unique feature of Lode Runner that gives it a rating of 9 out of 10.

LODE RUNNER from Broderbund  
48K disk (\$34.95)

## SAVE YOURSELF SOME TIME!

With *The Programmer's Toolkit* and Atari BASIC

**Write Programs Faster!** Thirty-one powerful routines are at your fingertips — things like decimal alignment and kid-proof keyboard entry. You can even play background music while your program executes!

**Write Faster Programs!** Use the speed of machine language for disk file reads and writes, string searches, text justification, and player-missile graphics. And more!

**Learn Fast, Too!** You don't need to learn a new language — it's all done with BASIC. A user's manual guides you through forty-two diskette examples — no lengthy typing!

**Time is Money — but just a little!** Thirty-one routines for \$29.95 — that's less than a dollar each! A free loan analysis program is included.

**The Programmer's Toolkit \$29.95**

(on diskette for Atari 400/800 with 24K RAM)

At Your Dealer or from:

SureSoft (918) 743-5363  
8177 S. Harvard, Suite 428  
Tulsa, Oklahoma 74137

**Sure  
Soft**

Add \$2 shipping and handling. VISA and MasterCard accepted by phone. OK residents add 5% sales tax. Allow two weeks for personal checks.

Dealer inquiries are invited.

Atari is a trademark of Atari, Inc. The Programmer's Toolkit is a trademark of SureSoft.



TID BITSCoin Program

by Lora &amp; Steve Van Slyke - JACG

Have you ever wanted to keep track of your coin collection by the date? This program was designed to do just that. It's menu supplies the user with options of Input, Display, Save, Retrieve, and End the program. Starting with the input; it will ask you for the date of the coin; however, it is limited to the dates of 1900 to 1999 for inputs. If you have finished inputting the dates just press <RETURN> twice after last date and you will proceed to the menu. You can then save your data on a disk, or retrieve data if you wish to expand your collection. In the display mode, you will get a number of how many coins you have on a certain date. Columns by decades, rows by single years and then at the bottom of the screen you will get the total accumulated coins.

The count of coins is in a single subscript array called P (line 70). Also the menu response (lines 162 and 350) is checked for other inputs besides those used. The display takes a new approach (lines 990-1210) because it prints out vertically instead of horizontally. It will also add and display the total coins in collection (lines 1230-1250). There is a menu (lines 156-162) that gets type of coins being used. A record check is built in, just in case there are no records to save (lines 680,685). There is another check in case the record is full (line 585).

If you have a cassette, instead of a disk, change line 70 to DIM P(99), F\$(2), CAT\$(8), X\$(1):F\$="C:" and delete line 163.

```

10 REM ***** COINS PROGRAM *****
20 REM BY LORA & STEVE VAN SLYKE
30 REM ** 4-2-83 VERSION 1.6 **
40 DIM P(99),F$(7),CAT$(8),X$(1):F$="D:-.DAT"
50 FOR D=0 TO 99:P(D)=0:NEXT D
60 REM ***** MENU *****
70 GRAPHICS 0
80 ? "Enter the coin category:"
90 ? ":?:?:?:?" 1.....Pennies"
100 ? " 2.....Nickels"
110 ? " 3.....Dimes"
120 ? " 4.....Quarters"
130 ? " 5.....Half Dollars"
140 L=155:TRAP 4000:INPUT C1:IF C1<1 OR C1>5 THEN GOTO 155
150 F$(3,3)=CHR$(64+C1)
160 FOR N=1 TO C1:READ CAT$:NEXT N
170 GRAPHICS 0
180 ? "Coin Collection Program"
190 ??:?:?:?"Your options are:"
200 ? " 1.....Display"
210 ? " 2.....Input"
220 ? " 3.....Save"
230 ? " 4.....Retrieve"

```

```

340 ? " 5.....End Program"
350 ??:?:?:?"What is your choice";
360 L=170:TRAP 4000:INPUT C:IF C<1 OR C>5 THEN 170
370 ON C GOTO 930,480,680,830
410 END
430 REM ***** INPUT *****
480 GRAPHICS 0
490 TRAP 610:?:?:CAT$:"What is the date, 19":INPUT D
570 IF D<0 OR D>99 THEN ? "Invalid input":GOTO 490
585 IF P(D)>254 THEN ? "The record is full for that date.":GOTO 490
590 P(D)=P(D)+1:GOTO 490
610 L=530:TRAP 40000:GOTO 170
650 REM ***** SAVE DATA *****
680 FOR X=0 TO 99:IF P(X)>0 THEN 690
685 NEXT X: ? "THERE ARE NO RECORDS":FOR G=1 TO 400:NEXT G:GOTO 170
690 OPEN #1,8,0,F$
710 FOR X=0 TO 99
730 A=P(X):PUT #1,A
750 NEXT X:CLOSE #1:GOTO 170
790 REM ***** RETRIEVE DATA *****
830 TRAP 910
850 OPEN #1,4,0,F$
870 FOR X=0 TO 99
890 GET #1,A:P(X)=A:NEXT X
910 CLOSE #1:TRAP 40000:GOTO 170
950 REM ***** DISPLAY *****
990 GRAPHICS 0:TD=0
1005 X=INT((20-(LEN(CAT$)/2))):FOR Y=1 TO X: ? " ";:NEXT Y: ? CAT$
1010 TC=-20
1030 FOR C=3 TO 30 STEP 3:POKE 82,C:POKE 84,2:TC=TC+10
1050 IF SGN(TC)=-1 THEN ? " ":GOTO 1130
1070 IF SGN(TC)=0 THEN ? TC:TC:GOTO 1130
1110 ? TC
1130 FOR U=0 TO 9
1150 IF SGN(TC)=-1 THEN ? U:GOTO 1190
1170 ? P(TC+U)
1190 NEXT U:POKE 84,0
1210 NEXT C
1230 FOR X=0 TO 99:TD=TD+P(X):NEXT X
1250 POKE 82,0:POSITION 0,15: ? " TOTAL COINS:":TD:?:?:?"
PRESS <RETURN> TO GO BACK TO THE MENU":INPUT X*:GOTO 170
2000 DATA PENNIES,NICKELS,DIMES,QUARTERS,HALFS
4000 ? "INVALID INPUT TRY AGAIN ":FOR D=1 TO 400:NEXT D:GOTO L

```



## SUNNYVALE SECRETS

by the Secret Sunnyvale Correspondent

Rumors abound but the Secret Sunnyvale Correspondent knows all! I have the answers to some of your questions. Please contact this publication if YOUR question has not been answered! Due to ATARI's management shuffle, I have been out of touch with all of you. So, to keep things short, here's a list of what's new:

Q: What has happened to ATARI hardware?

A: While Ray Kassar was in charge of development, he thought the ATARI computers were nothing more than sophisticated 'game machines'! It was during this time that many of ATARI's real talent left to form their own companies. ATARI is now recovering with the new management structure.

Q: Can you tell me anything about the new Touch Tablet?

A: Yes, the same people (Island Software & David Thornberg) who developed the KOALA TOUCHPAD have also developed our Touch Tablet. The software is 100% compatible! This should be a good standard as many new programs (like Music Construction Set) are including a 'touchpad' option on the list of inputs (like paddle, keyboard & joystick). Also, ATARI's Touch Tablet has the stylus electrically connected to the tablet so that control is now on the stylus and not the pad! This should make some things easier.

Q: What ever happened to the North American Phillips takeover?

A: There was never any takeover planned! (This was a rumor out of control!) North American Phillips (NORELCO) was interested ONLY in the record business of Warner Communications! NOT ATARI computers!

Q: What new products are on the drawing boards now?

A: A new addition to the XL series using a 6509 chip will give ATARI access to 128K of memory. No release date on this one. The CP/M machine is up and running but, again, no date is set for introduction. The same is true for the universal interface box (to run ANY manufacturer's software - too bad). The 1030 modem has finally filled the pipelines and you should start seeing them soon! The 850 interface will be back in FULL production as you read this. Unfortunately, ATARI may be a little too late on this one - other manufactures have developed products that perform as if the 850 were installed!

Q: Where's DOS 3.0 ?!!!!

A: It HAS been released and should be on the shelves by the time you read this. If a friend of yours has 3.0, you may not want his version - There were many last-minute revisions!

That's it for this month! Keep your cards & letters coming.

Signed,

SSC

## TECH TALK

by John Baum, STS Video

This is a question and answer column in which your questions about the technical aspects of your Atari will be answered each month. If you have a service or technical question, please send it to CURRENT NOTES care of Tech Talk.





FINDIT

by Bill Schadt

When developing or debugging a long program, I often find that I stop and write a short utility program to accomplish some task. Often, I don't bother to store the utility program because it is either too short or I simply cannot be bothered. The FINDIT program listed below is, however, an exception because I use it so often.

Suppose that you have LISTed a program called PNAME to the disk, and you want to see every line in PNAME which contains something like the word "POKE", or "RESTORE" or any other character string. List the latest version of PNAME to the disk, and then simply RUN the FINDIT program. If your listed program is not called PNAME, simply put the correct name into line 10 of FINDIT.

When FINDIT is RUN, you are prompted with "FIND WHAT?". You type in the string (called FIND\$) which you want to find. FINDIT then opens the PNAME file on the disk and begins the process of testing each line to determine if the line contains FIND\$. If a match is found the entire line is printed by line 260. If no match is found, processing continues until the end of the PNAME file triggers the TRAP 400 set in line 30.

The result of the scanning process can be sent to the printer simply by replacing "?" in line 260 with LPRINT.

FINDIT is definitely not fast. It is a great program to run while you go get a beer or attend to some other necessary human function. In any case, it can be a handy utility during long program debugging sessions.

Notice the INPUT on line 15 and the content of line 16. I discovered that the program would not work if I typed one or more spaces in response to the "FIND WHAT?" prompt. I really don't know why the input of a space(s) does not work, but it does not. I hope that one of the hackers in AURA will explain it to me.

## \*\*\*\*\* PROGRAM LISTING \*\*\*\*\*

```

3 DIM N$(15),X$(300),FIND$(40)
6 REM N$=NAME OF PROGRAM TO BE
7 REM EXAMINED TO DETERMINE WHICH
8 REM LINES CONTAIN FIND$
10 N$="D:PGNAME."
15 ? "FIND WHAT?";INPUT FIND$
16 IF FIND$="" THEN FIND$=" "
20 OPEN #1,4,0,N$
25 LF=LEN(FIND$)-1
30 TRAP 400:REM CATCH ERROR 136 - EOF
100 INPUT #1;X$:REM READ NEXT LINE
110 REM DOES X$ CONTAIN FIND$ ?
200 FOR C=1 TO (LEN(X$)-LF)
220 IF X$(C,C+LF)=FIND$ THEN 260
230 NEXT C
240 GOTO 100
250 REM MATCH FOUND HERE
260 ? X$:POP :GOTO 100
400 CLOSE #1

```

Computer Browsing

by Mike Barrett

If you have a modem, by calling 873-0874, you can hook up to Tyson's Corner Center's shopping information service. This service, which is provided by ITSS, Inc. of Rockville, MD, gives you useful information about "what's happening" at the center, including films and performance times at the nine cinemas and sales and specials available from the various merchants.

You are presented with fifteen minutes to browse through various menus and to select items of interest. You are periodically reminded of your time remaining throughout your session, but since there is no charge for the service and don't have to identify yourself, you can simply call again if you run out of time.

One of the most interesting services available is "gift ideas," which enables you to get eight or nine suggestions for gifts according to the sex, age and lifestyle of the recipient and the price range of the gift. The results are not particularly sophisticated, but it is fun to use and you do get the name of the item, its price and the store where it is available.

According to the Journal newspapers, the Tyson's Corner service is the first of its kind in the nation for home computer use. A similar service is available at the center itself but, as the Journal points out, color graphics that appear there will not appear on the home screens. The excuse given is that most computer owners don't have the same graphics capability on their home computers and that those that do have incompatible equipment. Of course, if everyone owned an Atari, this would not be a problem.

I note that it never occurred to me to write this article until Art Corte, our program chairman, suggested it to me. I encourage other to do the same or, if they do not have the inclination, to let me know about similarly available services and I will try them out (barring excessive telephone charges) and inform the group. My number in Reston, VA is 437-7522 or at work (DC) 633-2148.

INDEX TO ADVERTISERS

Company	page
Applied Computer Associates . . . . .	BC
Bits 'n' Bytes. . . . .	BC
Consumer Electronics. . . . .	2
RenumberIt. . . . .	10
Robins. . . . .	22
Bennet Rutledge, CDP. . . . .	19
Sure Soft . . . . .	24



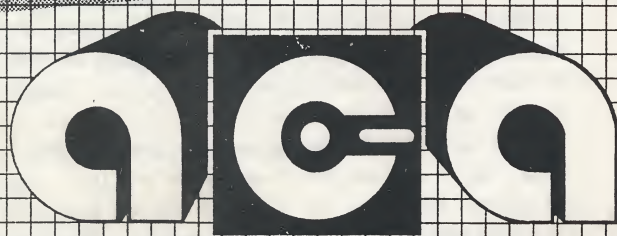
# "Washington's Leading ATARI Dealer"

creators of THE SOFTWARE CLUB™ and featuring the full line of ATARI brand products and

- MOSAIC
- MICROBITS
- EPSON-PROWRITER
- OKIDATA
- RANA-ASTRA

- PANASONIC
- BROTHER/COMREX
- BOOKS & ACCESSORIES
- HAPPY 810 ENHANCEMENT
- SYNAPSE, INFOCOM, MMG, etc.

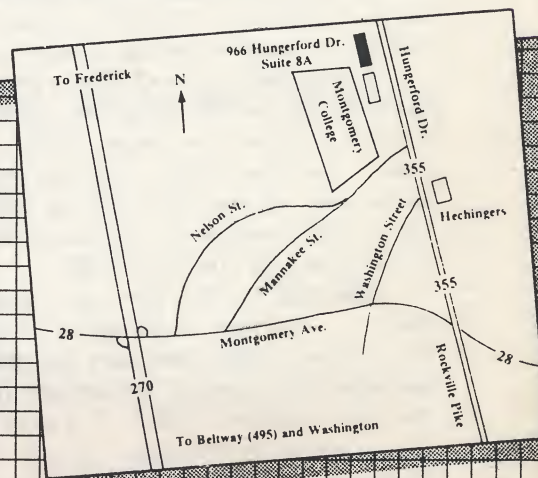
AND MANY, MANY MORE



Applied Computer Associates  
966 Hungerford Drive, #8A, Rockville, MD

(301) 424-4112  
Mon.-Wed. 10-6

**340-0100 or 424-4112**



Beginning Nov. 14th  
Thurs.-Sat. 10-8 Sun. 12-5

## BITS "n" BYTES

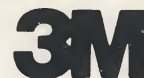
COMPUTER SUPPLIES  
Serving The Entire Metropolitan Area

- |                      |                         |                             |
|----------------------|-------------------------|-----------------------------|
| ■ Magnetic Media     | ■ Ribbons               | ■ Thermal Paper             |
| ■ Computer Paper     | ■ Custom Computer Forms | ■ Computer Furniture        |
| ■ Computer LH & Env. | ■ Print Wheels          | ■ CRT Film                  |
| ■ Labels             | ■ Data Binders          | ■ Static Control Floor Mats |
| ■ Tab Index Cards    | ■ Computer Cables       | ■ Head Cleaning Kits        |

For Large EDP Centers, Mini Computer and Word Processing Users

A Tradition of  
Quality Products and Service.

Authorized Distributor  
Information Processing Products



A Division of  
World of Print

**340-3345**

620 Hungerford Dr.  
Rockville, Md.